Capítulo 1

Identidade Digital Descentralizada: Conceitos, aplicações, iniciativas, plataforma de desenvolvimento e implementação de caso de uso.

Emilio Tissato Nakamura[1], Fernando Cezar Herédia Marino[1], José Reynaldo Formigoni Filho[1], Sérgio Luís Ribeiro[1] e Vítor Padilha de Oliveira[1]

[1]CPQD
Rua Ricardo Benetton, 1000, Campinas, SP – Brasil

Abstract

This short course aims to demonstrate a conceptual and practical view of Decentralized Digital Identity of things and people based on blockchain. We will present the main concepts related to blockchain, digital identity and self-sovereign digital identity, global initiatives, as well as legal and standardization aspects. The hands-on part will include the creation of a DLT network specialized for creating and managing Decentralized Digital Identities as well as the development of an application that executes the main methods and routines of the DLT (Hyperledger Indy) framework, such as: issuing decentralized identifiers ("DIDs") between applications ("Agents"), managing these identifiers, proof authentication and revocation of credentials by the issuer.

Resumo

Esse minicurso tem o objetivo apresentar uma visão conceitual e prática de Identidade Digital Descentralizada, de coisas e pessoas baseada em blockchain. Para isso, serão apresentados os principais conceitos relacionados a blockchain, identidade digital e identidade digital autossoberana, iniciativas globais, assim como os aspectos legais e de padronização. A parte prática do minicurso contemplará a criação de uma rede DLT, especializada para a criação e gestão de Identidades Digitais Descentralizadas, assim como o desenvolvimento de uma aplicação que execute os principais métodos e rotinas do framework DLT (Hyperledger Indy), como por exemplo, emissão de identificadores descentralizados ("DIDs") entre aplicações ("Agents"), gerenciamento desses identificadores, autenticação de provas e a revogação de credenciais pelo seu emissor.

1.1 Introdução

Quando alcançamos a entrada do período Neolítico, há cerca de dez mil anos, os grupos humanos existentes já acumulavam um variado leque de saberes apreendidos graças à sua habilidade de raciocínio. Foi nesse contexto que uma profunda transformação passou a se desenvolver no cotidiano do homem pré-histórico, surgindo assim as primeiras técnicas de cultivo agrícola, garantindo alimento sem a necessidade de deslocamento, criando as primeiras comunidades. Essa transformação, que se difundiu ao longo dos próximos seis mil anos, deu origem à chamada "Revolução Neolítica ou Revolução Agrícola" [1].

Desde então, começamos a seguir e evoluir baseado no modelo centralizado – que está perpetuado em nossas mentes – onde, teoricamente, uma pessoa ou governo tomam decisões (por consenso centralizado) para beneficiar a comunidade e a população como um todo, nesse sentido a centralização tem sido um princípio organizacional básico para a economia e a sociedade desde então.

Organizações maiores com mais clientes tendem a usar integração vertical e funções centralizadas para aumentar a eficiência e reduzir os custos [2]. Observa-se que a centralização econômica é o princípio organizacional mais eficaz desde que, os custos de comunicação e transação sejam elevados[2]. Porém, notamos que esse modelo centralizado está mudando e trará inúmeras implicações para a humanidade como um todo. Pois, a Internet diminuiu os custos da comunicação, a qual tende a zero, e a tecnologia *blockchain* fará o mesmo para os custos da transação [3] quebrando assim, o paradigma do modelo centralizado de eficiência e redução de custos.

Acredita-se que a tecnologia *blockchain* tem o potencial de ser a força motora que irá democratizar a economia mundial, e com certeza, será considerada uma das tecnologias mais importante na história do século. Pois pelo modelo desenvolvido, nenhuma autoridade central é necessária, criando assim, a maior quebra de paradigma na qual precisamos nos habituar e entender, pois o método de consenso será decentralizado [4]. Além disso, a arquitetura de confiança empregada na tecnologia também é descentralizada, o que a torna mais eficiente, pois aproxima o cliente final e o vendedor sem intermediários. Nos processos centralizados, o modelo de confiança é baseado em Leis, o que normalmente cria barreiras e/ou proteção de mercado à entrada de novos participantes.

A premissa atual é que governo e tecnologia num futuro próximo serão uma coisa só e a atividade de governar passará necessariamente pelo uso da tecnologia. A consequência disso é que um governo que não evolui tecnologicamente deixa de ser governo tornando-se obsoleto e ineficaz. Alinhado a esse tipo de ação, temos dois exemplos, o primeiro é da Estônia, um país case no uso de tecnologia e principalmente identidade digital e *blockchain*.

A Estônia implementou um pioneiro sistema de identidades digitais [5], com o objetivo de fazer com que tudo o que o cidadão precise do Estado possa ser feito de forma digital. Uma pessoa só precisa ir pessoalmente a um órgão público em três casos: (i) casar, (ii) divorciar , e (iii) para vender um imóvel.

Outro exemplo é o da Índia, que também criou uma identidade digital – Aadhaar [6] para os seus 1,2 bilhões de habitantes. A identidade digital indiana se organiza em torno de quatro princípios: (i) dispensa de presença física, (ii) dispensa de papel, (iii) digitalização do dinheiro e consentimento.

Esses princípios significam que os indianos atualmente podem acessar todos os serviços públicos pelo celular e sem precisar de nenhum papel. Além disso, o sistema

protege a privacidade, permitindo que o cidadão decida qual dado pode ser acessado por qual órgão (e somente por ele). Tudo isso, como consequência, levou a um processo de intensa bancarização da população, permitindo a digitalização da economia e a redução da circulação física de dinheiro.

1.2 Conceitos da Tecnologia *Blockchain*

A tecnologia *blockchain* pode ser entendida de várias formas. Em linhas gerais, pode-se dizer que se trata de um sistema distribuído de base de dados em log, mantido e gerido de forma compartilhada e descentralizada (através de uma rede *peer-to-peer* - P2P), na qual todos os participantes são responsáveis por armazenar e manter a base de dados.

A tecnologia foi construída tendo em mente quatro principais características arquiteturais: (i) segurança das operações, (ii) descentralização de armazenamento e computação, (iii) integridade de dados e (iv) imutabilidade de transações.

Dito de outra forma, *blockchain* é uma "*ledger of facts*" replicada em computadores que participam de uma rede *peer-to-peer*, onde [7]:

- O *ledger* é um livro de registros digital, no qual uma vez validado um registro, este nunca mais poderá ser apagado;
- Um fato (*fact*) pode significar várias coisas, desde uma transação monetária, a um conteúdo de determinado documento, ou até mesmo um programa de computador, contendo, em algumas plataformas, até uma base de dados pequena;
- Os membros participantes da rede podem, ou não ser anônimos e são chamados *peers* ou "nós";
- Toda operação ou transação dentro da *ledger* é protegida por tecnologias criptográficas de assinatura digital, inclusive para identificar os nós emissores e receptores das transações;
- Quando um nó deseja adicionar ao *ledger* um fato novo, é necessário um consenso entre todos ou alguns nós previamente determinados da rede, para decidir se um fato pode ser registrado no *ledger*;
- Havendo consenso, o fato será escrito e nunca mais poderá ser apagado, em tese, um processo levemente semelhante à escritura e registro de um imóvel no Brasil.

Conforme apresentado na Figura 1, uma rede *blockchain* possui os seguintes elementos essenciais:

- Fato (*Fact*): pode ser uma transação, um conteúdo digital ou um programa de computador, este último também denominado contrato inteligente (*smart contract*);
- Bloco: é um conjunto de fatos, geralmente em um número fixo predefinido;
- Cadeia de blocos (*blockchain*): conjunto de blocos encadeados (conectados um a um) seguindo uma lógica matemática, ou seja, não são independentes.

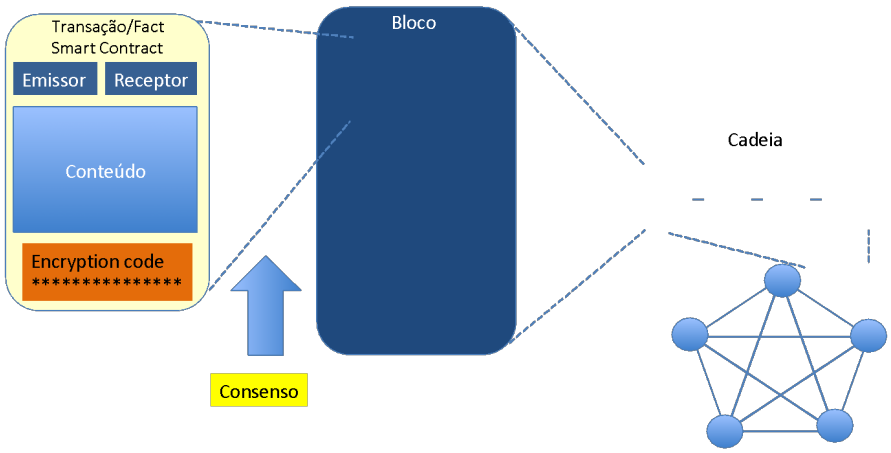
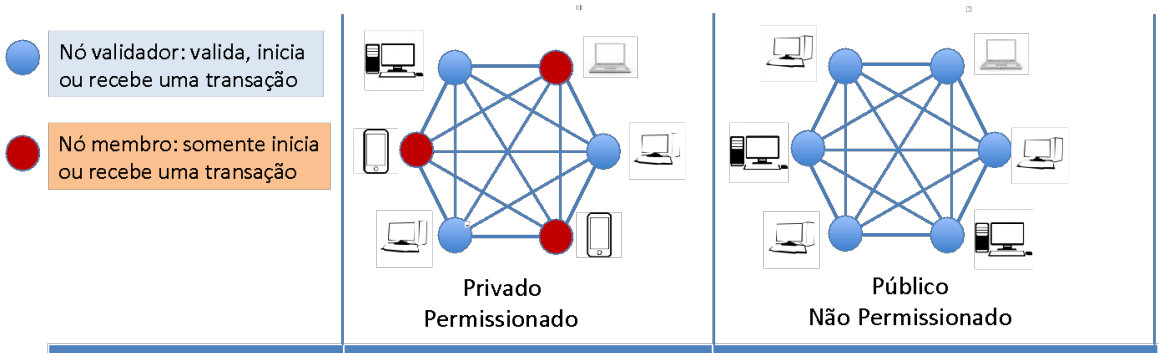


Figura 1. Fato, bloco e cadeia de blocos. Adaptado de [8].

Do ponto de vista de aplicação, a tecnologia *blockchain* passou por uma grande evolução com a possibilidade de uso dos contratos inteligentes que são programas de computador replicados e executados por todos os nós da rede, ou por um conjunto predeterminado de nós denominados validadores. Aplicações baseadas em contratos inteligentes são chamadas *Decentralized Applications* ou Dapps.

Atualmente, as redes *blockchain* são divididas em dois grandes grupos Figura 2:

- ***Blockchains* públicas**: Os protocolos de *blockchain* públicos de última geração são baseados em algoritmos de consenso conhecido como *proof of work* (PoW) são de código aberto e acesso aberto (*permissionless*), qualquer um pode participar sem permissão.
    - Qualquer um pode baixar o código e começar a executar um nó público em seu dispositivo local, validando transações na rede, participando do processo de consenso.
    - Qualquer pessoa no mundo pode enviar transações através da rede e esperar vê-las incluídas no *blockchain* se elas forem válidas.
    - Qualquer pessoa pode ler transações no explorador público de blocos. As transações são transparentes, porém são anônimas ou pseudoanônimas.
- ***Blockchains* privadas**: São as redes de acesso autorizado (*permissioned*). As permissões de gravação são mantidas centralizadas em uma organização. Permissões de leitura podem ser públicas ou restritas a uma extensão arbitrária. Exemplos de aplicativos incluem gerenciamento de banco de dados, auditoria, etc., que são internos a uma única empresa e, portanto, a legibilidade pública pode, em muitos casos, não ser necessária. Os *blockchains* privados são uma maneira de aproveitar a tecnologia *blockchain*, configurando grupos e participantes que podem verificar as transações internamente, essa estratégia pode criar alguns riscos de violação, pois o sistema é centralizado. No entanto, *blockchains* privados têm inúmeros casos de uso, especialmente quando se trata de escalabilidade, conformidade do estado das regras, e privacidade de dados, além de outros problemas de regulamentação.



Acesso à rede	Necessita de autorização	Acesso aberto
Aspectos legais e regulatórios	Alinhado	Não considera, tem regras próprias
Quem são os nós validadores	Grupo pré selecionado	Anônimos
Potenciais aplicações	Ambientes corporativos fechados	Aplicações de acesso aberto (Bitcoin, por exemplo)

Figura 2. Redes *blockchain* privadas e públicas.

Uma outra visão interessante sobre a diferença entre esse dois grandes grupos, pode ser visto na Figura 3 o qual inclui o conceito de federação ou consórcio, *blockchains* federados operam sob a liderança de um grupo. Ao contrário dos *blockchains* públicos, eles não permitem que qualquer pessoa participe do processo de verificação de transações. Os *blockchains* federados são mais rápidos (maior escalabilidade) e fornecem mais privacidade de transações. O nome consórcio normalmente são usados no setor bancário. O processo de consenso é controlado por um conjunto pré-selecionado de nós, por exemplo, pode-se imaginar um consórcio de 15 instituições financeiras, cada uma operando um nó e das quais 10 devem assinar cada bloco para que o bloco seja válido. O direito de ler o *blockchain* pode ser público ou restrito aos participantes.

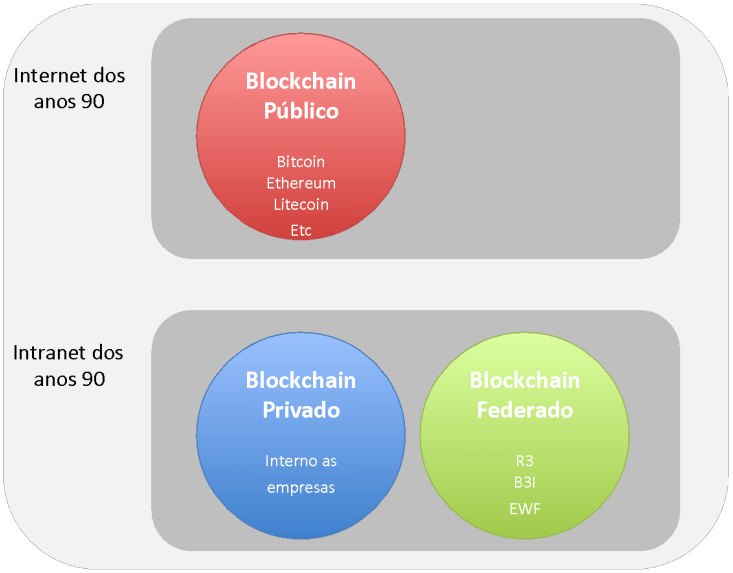


Figura 3. Redes *blockchain* privadas, públicas e federadas. Adaptado de [9].

1.3 Tecnologia *Blockchain* e a Segurança

Apesar dos problemas de segurança, divulgados, utilizando a tecnologia *blockchain*, seja na operação de criptomoedas ou em iniciativas como da DAO [10][11], vale ressaltar que os ataques foram direcionadas as aplicações que utilizam o *blockchain*, e não especificamente a tecnologia ou ainda ao algoritmo empregado.

Além disso, observa-se que os ataques bem-sucedidos, relatados até o momento, a plataformas baseadas em *blockchain*, como por exemplo Bitcoin [12], ocorreram devido a vulnerabilidades nas aplicações e não no core da tecnologia *blockchain* propriamente dito. Sendo assim, com relação ao aspecto de segurança, até o presente momento, não são conhecidas vulnerabilidades contra a construção empregada e algoritmos utilizados nativamente no *blockchain*, podendo assim dizer que segurança ainda é um dos pontos fortes da solução.

Para que isso ocorra o algoritmo prevê que no processo de inserção de novos blocos, um novo bloco, composto por um conjunto de transações, é ligado criptograficamente aos blocos anteriores por meio de um processo chamado de validação. Que nos casos específicos de criptomoedas, Bitcoin por exemplo, é conhecido como mineração. O processo é computacionalmente intensivo e é o que faz com que seja improvável que modificações maliciosas possam ser realizadas por um atacante.

1.3.1 Camadas de Segurança de uma Plataforma *Blockchain*

A plataforma *blockchain* e as aplicações construídas com ele devem adotar a segurança em camadas. Há seis camadas de segurança a serem consideradas. Estas camadas são o resultado da compilação de boas práticas presentes na área de segurança da informação e são apresentadas na Figura 4 e são descritas a seguir [13].

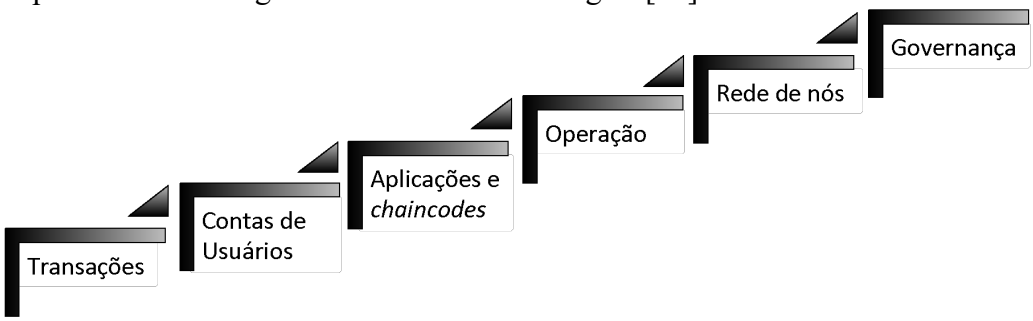


Figura 4. Camadas de segurança para um desenvolvimento *blockchain*.

- A camada fundamental e a primeira camada a ser considerada é a **segurança da transação**. Requisito mínimo sem o qual o *blockchain* não faz sentido. O *blockchain* deve validar as transações com confiança e previsibilidade ao final do consenso. O consenso vai confirmar a finalidade e a imutabilidade de transação.
  Trata-se de proteções sintática e estrutural para as transações e os blocos que as contém. Estas proteções não impedem fraudes semânticas associadas à lógica da aplicação;
- A segunda camada oferece **segurança da conta de usuário**. A conta do usuário é geralmente gerenciada pelo próprio usuário em aplicativos de uso pessoal (*eWallets*). Muitas vezes, a proteção da conta do usuário é confundida com a segurança do software cliente.

Esta camada de segurança é influenciada por dois fatores: a conscientização dos usuários no uso seguro da tecnologia, e a implementação correta dos mecanismos de segurança para dispositivos móveis e sistemas web.

- A terceira camada contempla a **segurança da aplicação e dos *chaincodes***. Fazem parte desta camada as boas práticas de desenvolvimento seguro de software, incluindo a codificação segura de *smart contracts* e a definição de requisitos de segurança, avaliação de arquitetura e testes de segurança da aplicação.
- A quarta camada atende a **segurança de implantação e de operação da aplicação**. Fazem parte desta camada os testes de aceitação e homologação da aplicação e dos *chaincodes* antes de implantação em produção. Uma vez no ambiente de produção, a aplicação deve ser monitorada para detecção de anomalias de funcionamento e comportamento. Monitoramentos avançados podem até detectar fraudes.
- A quinta camada cobre a **segurança da rede P2P de seus nós**. Nesta camada, os mecanismos de proteção tradicionais das redes de computadores (tais como sistemas de *firewall*, IDS, IPS, etc.) podem ser aplicados para proteção dos nós da rede P2P do *blockchain*. Além disso, proteções específicas devem ser aplicadas para a segurança do protocolo de comunicação e de consenso. Ainda, deve ser observada a quantidade mínima necessária de nós disponíveis para garantir o consenso.
- A sexta camada de segurança se refere à **governança da aplicação e do *blockchain***. Esta camada abriga aquelas decisões sobre a estrutura e projeto do *blockchain*, que afetam o funcionamento com segurança, incluindo ainda controles antifraude, auditoria, privacidade e até conformidade a normas padrões específicos do nicho de aplicação.

1.4 *Blockchain* e IoT

Temos que um dos problemas associado à Internet das Coisas (*Internet of Things* – IoT) são os aspectos relacionados a segurança e privacidade, fato esse que alcançou um novo patamar quando a rede de *bots* Mirai causou uma interrupção maciça na Internet em setembro de 2016 [14]. A vulnerabilidade explorada pelo Mirai foi no uso de um ataque baseado em senhas de dicionário em um número, sem precedentes, de dispositivos os quais ofereciam acesso direto à Internet [15][16].

Porém, pesquisas apontam que a utilização da tecnologia *blockchain* pode contribuir para a mitigação deste problema como já discutido em várias publicações dentre elas [17][18][19] onde alguns dos benefícios mais diretos na utilização da tecnologia são:

- Rastrear a história única de cada dispositivo, registrando a troca de dados com outros dispositivos, serviços web e usuários humanos;
- Permitir que dispositivos inteligentes atuem de forma autônoma em uma variedade de transações;
- Monitoramento remoto de ativos de elevado valor para verificar, por exemplo, se estão sendo usados corretamente;

- Monitoramento, controle e autorização de solicitação de determinado equipamento para reposição de alguma peça ou matéria-prima (máquina de lavar solicitando sabão, por exemplo);
- Controle de identidade dos dispositivos IoT para registro e controle de acesso lógico a diferentes aplicações.

Outros exemplos de aplicação envolvendo mais de um setor também podem ser pensados. A tecnologia *blockchain* pode ser aplicada em projetos estruturantes, que envolveriam diferentes atores de uma cadeia de valor, tais como:

- Monitoramento e rastreamento de uma cadeia de produção (p.ex.: fabricação de automóveis, produção de vinhos, produção de equipamentos de informática, dentre outros);
- Sistema de gestão de logística reversa de diferentes produtos (p.ex.: produção de medicamentos, produtos eletroeletrônicos e seus resíduos); e
- Sistemas de gestão e controle da distribuição de venda de produtos sob regime regulatório forte (p.ex.: medicamentos de uso controlado, procedência de carne e alimentos orgânicos).

Em suma, as aplicações para IoT tem potencial para usufruir dos benefícios que a tecnologia *blockchain* traz consigo [20], resumidos na **Tabela 1**.

Tabela 1: Benefícios potenciais da *blockchain* em IoT.

Benefício	Descrição
Transações confiáveis, rápidas e sem intermediário	Reduz ou mesmo elimina o risco da desconfiança entre as partes e custos de transação.
Usuários com poderes	Usuário controla todas as suas transações e informações
Dados de alta qualidade	Os dados da *blockchain* intrinsecamente são completos, consistente, precisos e amplamente disponíveis no momento que forem necessários.
Duráveis, confiáveis e amplamente disponíveis	Ausência de ponto único de falha.
Processos íntegros	Confiança que tudo será executado conforme as regras pré-definidas sem intermediários.
Transparência e imutabilidade	Todas as transações podem estar disponíveis publicamente e não podem ser alteradas ou ainda apagadas dos registros.
Simplificação do ecossistema	Um único livro de registro é criado, reduzindo a desordem e complicações.

À medida que o número de dispositivos conectados cresce de milhões para bilhões, e governos e corporações correm para melhor controlar dispositivos e dados, uma nova estratégia tecnológica será necessária para construir soluções de baixo custo que levem em consideração privacidade e autonomia.

Novos modelos de negócio guiarão estas soluções na direção de economias digitais eficientes e na criação de valor de forma colaborativa, ao mesmo tempo em que serão criados experiência de usuários e produtos melhorados.

No nível mais abstrato, as próprias redes podem se tornar autônomas suplantando sistemas já estabelecidos e que hoje dependem de uma autoridade centralizadora, como, por exemplo, a troca de informação sensível e serviços de auto-instalação e *auto-update* de *software* em dispositivos [21]. Nesse sentido, qualquer

solução IoT descentralizada deveria suportar: (i) mensagens P2P confiáveis, (ii) comunicação intrinsecamente confiável e (iii) autonomia descentralizada.

Em linha com essa perspectiva evolutiva da IoT, uma Cidade Inteligente é um bom exemplo de como combinar IoT e *blockchain*. Os serviços de compartilhamento baseados em *blockchain* podem contribuir para as cidades inteligentes.

A economia compartilhada, nesse caso, é um modelo econômico-social que diversos setores da população podem utilizar para fazer uso compartilhado de ativos subutilizados [22]. Os cidadãos, objetos e *utilities* se conectariam de forma transparente para compartilhar o status e a troca desses ativos.

Nesse paradigma, as pessoas buscam confiança, ter acesso ao invés de ter propriedade, confiabilidade dos serviços compartilhados e segurança e privacidade.

As instituições precisam de uma comunidade inteligente com governo inteligente, parcerias, networking e governança transparentes, interconexão dinâmica com as partes interessadas e estar protegidas de fraudes, responsabilidades e prestadores de serviço desqualificados.

A computação precisa garantir acessibilidade e disponibilidade dos sistemas, recursos de bases de dados inteligentes; sistemas de controle; interface, computação inteligente; habitantes em rede e ciência em tempo real e analíticos avançados.

Há seis elementos que *blockchain* ajudaria nos relacionamentos entre pessoas, tecnologia e organizações: (i) não depender da confiança entre os atores, (ii) transparência e privacidade, (iii) democratização, (iv) automação, (v) ser distribuído e (vi) segurança [23].

1.5 Identidade Digital

As aplicações relacionadas a identidade digital utilizando tecnologia *blockchain* permitem a verificação, autorização e gerenciamento de identidade inalterados, resultando em eficiências significativas e redução de fraudes.

A tecnologia *blockchain* fornece o mecanismo ideal para identidades digitais. Enquanto as identidades digitais estão emergindo como uma parte inevitável do nosso mundo conectado, a forma como protegemos nossas informações on-line está sendo submetida a um intenso escrutínio. Os sistemas de identidade baseados em *blockchain* podem fornecer uma solução para esse problema com criptografia rígida e *ledgers* distribuídos.

Os recentes casos de violações de dados, vazamentos e uso indevido dominaram as manchetes ao longo do ano passado, trazendo nova proeminência às questões de proteção de dados pessoais. O escândalo do Facebook-Cambridge Analytica, a violação de dados do provedor LocationSmart e outros levaram os usuários e os reguladores a examinarem mais de perto como as empresas privadas estão lucrando - e às vezes abusando - dos dados de identidade do cliente. No campo da tecnologia *blockchain* para identidade, vimos as empresas reagirem com um modelo de negócios relativamente novo: o mercado de identidades pessoais [24].

Um pequeno mas crescente contingente de empresas e principalmente *startups* estão desenvolvendo serviços para mudar a monetização de dados pessoais de empresas digitais e anunciantes para os próprios usuários, que são os reais donos da identidade. Esses players dão aos consumidores mais controle sobre como eles “usam” seus dados, combinando funções de identidade e criptomoeda, de forma que os usuários são compensados por atributos individuais que escolhem compartilhar com empresas

privadas. Startups como Datum, DataVest e Wibson, por exemplo, surgiram em 2018 com base nessa funcionalidade. Ainda outras empresas como Civic e Procivis, planejam lançar um novo mercado de *token* totalmente voltado para transações de dados pessoais [24].

Mercados de dados pessoais baseados em *blockchain* são uma proposta intrigante, mas exigem que uma base de consumidores particularmente importante, motivada e digitalmente experiente utilize e consequentemente mude significativamente a economia de dados pessoais.

Até o momento, são poucos os usuários que demonstram interesse em adotar serviços relacionados a privacidade ou o controle adicional sobre suas identidades digitais, portanto os mercados de dados pessoais devem contar com uma estrutura de incentivos forte e experiência de usuário intuitiva para atrair esse público.

1.6 Identidade Digital Autossoberana

Mais de vinte e cinco anos se passaram desde que Peter Steiner mostrou ao mundo pela primeira vez que "Na Internet, ninguém sabe que você é um cachorro" [25], mas esse famoso *cartoon* ainda continua atual e válido, pois representa o desafio de identificar pessoas on-line.

Hoje, estamos muito longe da visão de diretórios públicos, a qual era a expectativa da criptografia de chave pública nos anos setenta ou do grande esquema de certificação hierárquica previsto nos anos oitenta. O gerenciamento de identidades (IdM) na Internet ainda conta com o que Cameron [26], a mais de uma década, chamou de uma "*colcha de retalhos de identidades únicas*", compreendendo vários tipos de sistemas IdM que são restritos a domínios específicos e não interagem entre si.

Os modelos centralizados de IdM enfrentam atualmente inúmeros desafios devido à crescente legislação relacionado às violações de dados, que levam a danos à reputação, fraude de identidade, mas acima de tudo uma perda de privacidade para todos os envolvidos. Esses eventos recorrentes destacam a falta de controle e gestão que os usuários experimentam com suas identidades digitais [27][28][29].

A pesquisa de abordagens alternativas à IdM está sendo conduzida por iniciativas que buscam ampliar a confiabilidade e o alcance das formas digitais de identidade. A Estratégia Nacional dos Estados Unidos para Identidades Confiáveis no Ciberespaço (NSTIC) visa acelerar o desenvolvimento de novas tecnologias que podem aumentar a confiança nas transações on-line [30]. Além disso, o ID2020 procura alavancar tecnologias digitais emergentes para expandir o alcance de identidades legais (espelhando as metas das Nações Unidas de "fornecer até 2030 identidade digital para todos, incluindo o registro de nascimento" [31]). O surgimento do Bitcoin [32] também inspirou um novo pensamento sobre a identidade digital, devido à sua subjacente tecnologia de contabilidade distribuída (DLT), que não precisa de uma autoridade central para validar as transações de sua criptografia natural.

Assim, uma rede globalmente descentralizada é capaz de chegar a um consenso sobre o estado atual das transações. Dado que o DLT é adequado para assegurar o consenso, a transparência e a integridade das transações que ele contém, vários benefícios da aplicação do DLT ao IdM já foram propostos:

- **Descentralizada** - As informações de identidade são referenciadas em um razão que nenhuma autoridade central possui ou controla.

- **Inviolável** - Atividades históricas no DLT não podem ser adulteradas e transparência é dada a todas as mudanças nesses dados.
- **Inclusivo** - Novas maneiras de se criar identidade do usuário podem ser concebidas para expandir o alcance de identidades legais e reduzir a exclusão.
- **Redução de custos** - As informações de identidade compartilhada podem levar à redução de custos para as partes confiáveis, juntamente com o potencial de reduzir o volume de informações pessoais que são replicadas em bancos de dados.
- **Controle de usuário** - Os usuários não podem perder o controle de seus identificadores digitais, mesmo se perderem o acesso aos serviços de um determinado provedor de identidade.

1.7 Identidade Digital Autossoberana Baseado em *Blockchain*

O Gerenciamento de Identidade (IdM) abrange os processos e políticas envolvidos no gerenciamento do ciclo de vida de atributos em identidades para um domínio particular [33].

A maioria dos modelos de IdM hoje é centralizado, onde uma única entidade controla todo o sistema. No entanto, as próprias identidades geradas podem ser federadas além de uma única organização, como quando os governos emitem carteiras de identidade nacionais.

Nos sistemas de identidade federada, os usuários podem usar informações de identidade estabelecidas em um domínio de segurança para acessar outro. Esquemas de *login* único, como o Facebook e Google, por exemplo.

O gerenciamento de identidade centrado no usuário coloca a administração e o controle das informações de identidade diretamente nas mãos dos indivíduos. Os exemplos incluem gerenciadores de senhas (por exemplo, 1Password, Less-Pass, entre outros) que gerenciam, de maneira segura, as diferentes credenciais nos sites da internet.

Apesar das diferentes abordagens, uma função que é fundamental para o IdM é a vinculação segura de um identificador único: um valor que distingue inequivocamente um usuário de outro em um mesmo domínio. E também, atributos (às vezes chamado de certificações ou declarações): direitos ou propriedades de um usuário como nome, idade, classificação de crédito etc.

Os primeiros passos tomados para adequar o uso de DLT para estabelecer um mapeamento de identificador seguro e descentralizado foram tomados no design do *Namecoin* que é um dos *fork* mais longínquos do Bitcoin. O *Namecoin* fornece um *namespace* legível, descentralizado e seguro para o domínio ".bit". Essa conquista contradizia a sabedoria convencional de que um sistema de nomenclatura exibindo todas as três características não poderia ser projetado [34]. *Blockstack* [29] ampliou o esquema do *Namecoin*, para criar uma infraestrutura de chave pública (PKI) descentralizada que registra ligações entre uma chave pública e um identificador legível.

Recentemente, surgiram vários modelos de identidade descentralizada que se estendem além da nomenclatura e visam fornecer um conjunto mais completo de funções de IdM. No entanto, até o momento, não houve uma avaliação direta e ampla dessas propostas.

Atualmente existem basicamente duas categorias de propostas de IdM baseado em *blockchain*:

1. ***Self-Sovereign Identity* (SSI):** uma identidade que pertence e é controlada por seu proprietário sem a necessidade de depender de qualquer autoridade administrativa externa e sem a possibilidade de que essa identidade possa ser removida. Ele pode ser ativado por um ecossistema de identidade descentralizado que facilita o registro e a troca de atributos de identidade e a propagação da confiança entre as entidades participantes. Exemplos incluem Sovrin, uPort e OneName;
2. **Identidade confiável descentralizada:** uma identidade fornecida por um serviço centralizado que realiza a prova de identidade de usuários com base em credenciais confiáveis existentes (por exemplo, passaporte) e registra atestados de identidade em um DLT para validação posterior por terceiros. Exemplos incluem ShoCard, BitID, ID.me e IDchainZ.

1.8 Iniciativas Globais em Identidade Digital Autossoberana

Como exemplo, será apresentado duas iniciativas globais baseadas em SSI, sendo as inciativas da Sovrin e da uPort, e também uma inciativa baseada em identidade confiável decentralizada, nesse caso o da ShoCard.

Acredita-se que com a apresentação dessas três iniciativas globais é possível identificar o modelo de atuação, decisões de design predominantes e também os desafios encontrados. Interessante notar que esses três exemplos servem a um propósito similar para o cenário mais amplo do IdM baseado em DLT.

Essas iniciativas foram escolhidas pois fornecem de forma clara os detalhes técnicos de seus projetos, além disso, são iniciativas sustentadas por comunidades de grande porte ou ainda têm notável nível de financiamento.

1.8.1 Leis da Identidade

Na seção de análise de cada exemplo (Seção 1.8.2.2), (Seção 1.8.3.2), e (Seção 1.8.4.2), serão brevemente analisados perante as conhecidas “Leis da Identidade” [26] que usualmente servem como parâmetros para identificar os sucessos e fracassos dos sistemas de identidade digital. Essas leis compõe uma estrutura conhecida e representa um espectro completo de preocupações com IdM, abrangendo segurança, privacidade e experiência do usuário, as 7 Leis da Identidade são descritas a seguir.

Lei 1 - Controle e consentimento do usuário - As informações que identificam o usuário só devem ser reveladas com o consentimento desse usuário.

Lei 2 - Divulgação mínima e para uso restrito - As informações de identidade só devem ser coletadas em uma base de “necessidade de conhecimento” e mantidas em uma base de “necessidade de reter”.

Lei 3 - Partes justificáveis - As informações de identidade só devem ser compartilhadas com partes que tenham direito legítimo de acessar informações de identidade em uma transação.

Lei 4 - Identidade dirigida - O suporte deve ser fornecido para compartilhar informações de identidade publicamente ou de maneira mais discreta.

Lei 5 - Design para um pluralismo de operadores e tecnologia - Uma solução deve permitir a interoperabilidade de diferentes esquemas de identidade e credenciais.

Lei 6 - Integração humana - A experiência do usuário deve ser consistente com as necessidades e expectativas para que os usuários possam entender as implicações de suas interações com o sistema.

Lei 7 - Experiência consistente em contextos - Os usuários devem ser capazes de esperar uma experiência consistente em diferentes contextos de segurança e plataformas de tecnologia.

1.8.2 *Self-Sovereign Identity* – Sovrin

Sovrin [27] é uma rede de identidade descentralizada de código aberto construída sobre a tecnologia DLT (Figura 5), é considerado uma rede pública / permissionada, onde, apenas as instituições confiáveis chamadas de *stweards* ou *writers* que podem ser bancos, universidades, governos, instituições de pesquisa, etc – são os nós que participam do consenso e executam a gravação na *ledger*. Já os nós observadores só possuem atributo de leitura da *ledger* e atuam como intermediários entre o usuário final e a rede.

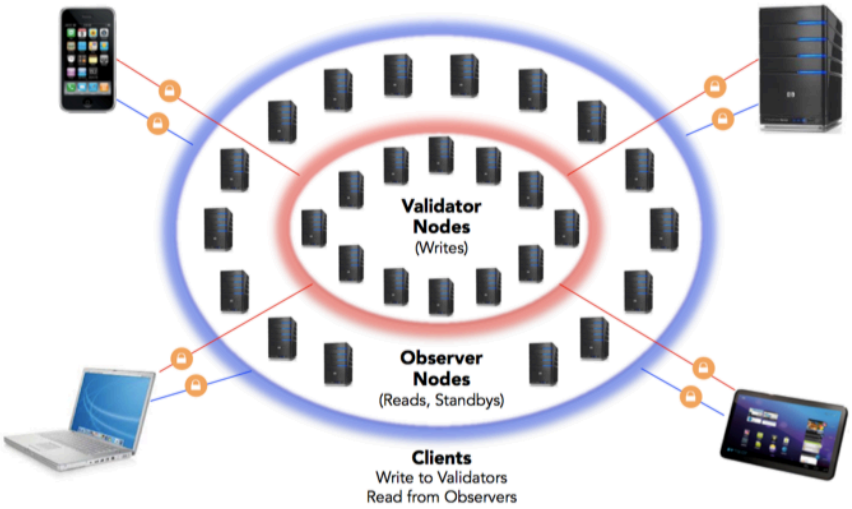


Figura 5. Sovrin: Nós validadores e nós observadores.

A Fundação Sovrin, sem fins lucrativos, assegura a governança adequada dos administradores e o respeito ao acordo legal denominado Sovrin *Trust Framework* firmado entre a fundação e os *stweards/writers*. A Fundação Sovrin é quem fornece o código-base para o projeto Hyperledger Indy [35].

1.8.2.1 Sovrin: Abordagem

A Sovrin permite que um usuário gere tantos identificadores quantos forem necessários para manter a separação contextual de identidades para fins de privacidade. Cada identificador é único, "*unlinkable*" e controlado por um par de chaves assimétricas distintas. Os identificadores da Sovrin são gerenciados pelo usuário ou por um serviço de intermediário designado (agente), e seguem a especificação de Identificador Descentralizado (DID) padrão W3C [36]. Um DID é uma estrutura de dados contendo o identificador do usuário, a chave pública criptográfica e outros meta-dados necessários para transacionar com esse identificador.

A arquitetura da Sovrin pode ser resumida pelos componentes, conforme mostrado na Figura 6. O elemento-chave é a *ledger* Sovrin a qual contém transações

associadas a um identificador específico e é, gravado, distribuído e replicado entre os nós *stewards*, que executam uma versão aprimorada do protocolo redundante de tolerância a falhas bizantina de Aublin et al. [37], chamado Plenum, para consenso [38].

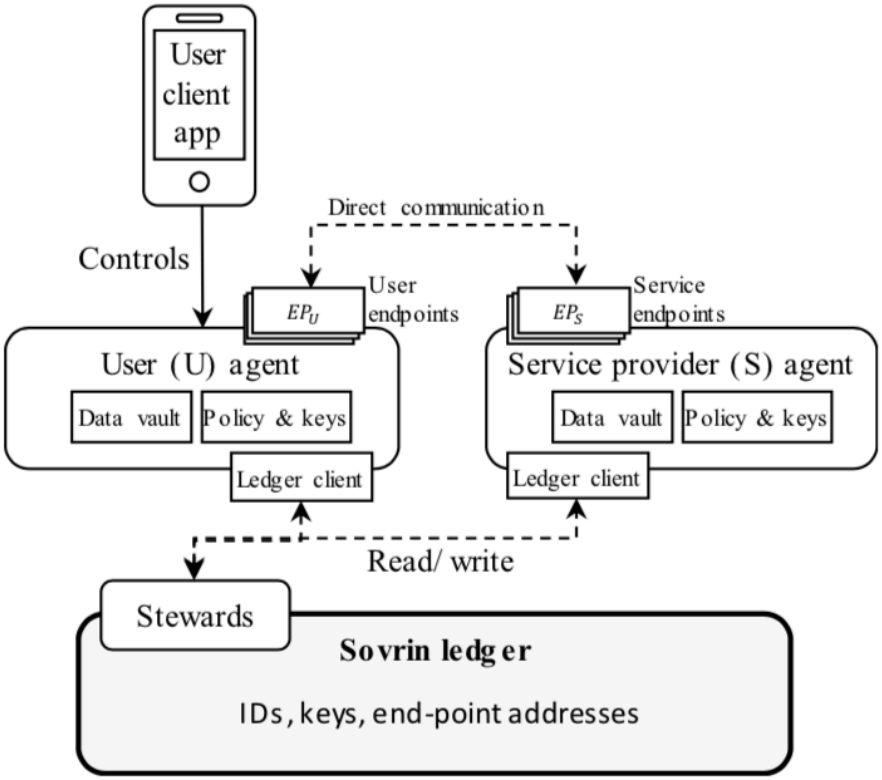


Figura 6. Sovrin: Elementos chave [38].

Há duas consequências importantes para a escolha do consenso ser permissionado no design da Sovrin. Primeiro, não é necessário nenhum cálculo "caro" para chegar a um consenso sobre o estado da *ledger*, reduzindo significativamente o custo de energia da execução de um nó e melhorando drasticamente a performance e tempo de resposta das transações. Em segundo lugar, a confiança na rede Sovrin reside tanto nas pessoas como no código. A confiança começa a partir da raiz comum de confiança formada pelo *ledger* distribuído globalmente, mas à medida que novas organizações e usuários ingressam na rede, eles podem se tornar âncoras de confiança (ou seja, podem adicionar mais usuários e organizações). Espera-se que uma "rede de confiança" evolua para apoiar esse crescimento descentralizado da rede.

Os usuários interagem com a Sovrin através de uma aplicação móvel e controlam os agentes de *software* que agem em seu nome para facilitar as interações com outros agentes na rede (Figura 7). Os *endpoints* de rede devem ser sempre endereçáveis e acessíveis. Os usuários/*endpoints* podem executar as ações em seus próprios servidores, mas provavelmente, a utilização se dará pelos intermediários (agentes) especializados. Os agentes também fornecem serviços de *backup* e armazenamento criptografado das credenciais por exemplo.

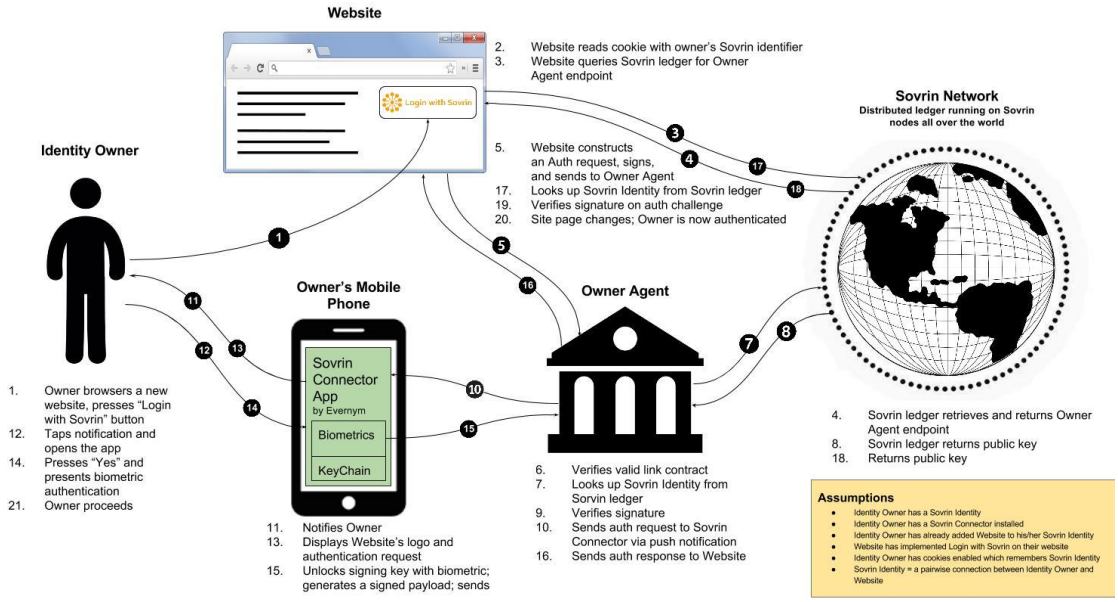


Figura 7. Sovrin arquitetura de comunicação [39].

1.8.2.2 Sovrin: Considerações Baseadas nas Leis da Identidade

A Sovrin visa fornecer aos usuários controle total a todos os aspectos da identidade. Cada usuário pode selecionar as credenciais de atributo que possui sobre si, que deseja compartilhar com uma terceira parte confiável **(Lei 1)**. Isso é possível através do uso de credenciais anônimas.

Embora os usuários possam optar por armazenar esses atributos na *ledger*, em geral, é recomendado que utilizem os recursos de armazenamento dos dispositivos móveis ou ainda recursos dos agentes para transmitir atributos a outras partes por meio de canais de comunicação seguros e usar a *ledger* para identificar a rede e *endpoint* a ser utilizado. O uso de credenciais baseadas em atributos permite que os usuários revelem apenas as credenciais que escolham **(Lei 2)**. A verificação da parte com quem os dados são partilhados continua a ser um desafio, que é parcialmente resolvido através da *web-of-trust*, da governança da Fundação Sovrin e da reputação dos *stewards*.

Embora não haja terceiros de confiança no sentido de PKI na Sovrin, os utilizadores devem confiar nas agências que "agem" em seu nome na rede da Sovrin e nos administradores responsáveis pela manutenção do *ledger*. Dependendo da escolha do agente e de sua implementação, muitas informações podem estar nas mãos desses agentes. No entanto, como as agentes agem em nome do usuário, eles têm um "lugar necessário e justificável" no relacionamento de identidade **(Lei 3)**.

A Sovrin suporta identificadores omnidirecionais e unidirecionais (Lei 4), pois as organizações públicas podem decidir publicar a sua identidade completa na rede, enquanto os utilizadores podem optar por publicar apenas identificadores e usar identificadores e pares de chaves criptográficas diferentes com cada parte com a qual interagem, evitando emitir "identificadores de correlação".

Atualmente a Sovrin ainda depende de um número reduzido de operadores distribuídos geograficamente. À medida que o sistema ganhar notoriedade, com certeza, novos agentes e novos *stewards* irão amentar essa capilaridade. A Fundação Sovrin espera, em particular, construir um mercado de agentes que competirão com as

características que oferecem, por exemplo, mecanismos de backup, carteiras, interfaces com outros sistemas de identidade existentes entre outros **(Lei 5)**.

Finalmente, uma questão importante ainda não amplamente abordada nem pela literatura nem pelos desenvolvedores da rede Sovrin é a experiência do usuário. O histórico de serviços de segurança oferece vários exemplos de sistemas criptográficos inteligentes, que nunca foram implantados amplamente porque os usuários acharam muito complicado ou difícil de entender - a criptografia de e-mail usando o PGP é um exemplo. Portanto, a integração humana continua sendo uma questão em aberto para a rede Sovrin. Considerando que apesar de todos esforço dedicado a arquitetura de sistema e do modelo de SSI empregado na Sovrin, a experiência do usuário não foi amplamente desenvolvido o que penaliza as Leis 6 **(Lei 6)** e 7 **(Lei 7)**.

1.8.3 Self-Sovereign Identity – uPort

O uPort [28] é uma estrutura de identidade descentralizada de código aberto que visa fornecer "identidade descentralizada para todos". Seu caso de uso é o IdM para aplicativos descentralizados de próxima geração (DApps) no Ethereum DLT [40] e para aplicativos tradicionais centralizados, como e-mail e bancos.

1.8.3.1 uPort Abordagem

Uma identidade da uPort é sustentada pelas interações entre os contratos inteligentes da Ethereum. Contratos inteligentes são endereçados exclusivamente por identificadores hexadecimais de 160 bits e, quando invocados, são executados pela Máquina Virtual Ethereum (EVM) instalada em cada nó Ethereum.

Dois modelos de contrato inteligentes projetados pela uPort compreendem cada identidade da uPort: controlador e proxy. Para criar uma nova identidade, o aplicativo móvel uPort de um usuário cria um novo par de chaves assimétricas e envia uma transação para a Ethereum que cria uma instanciação de um controlador que contém uma referência à chave pública recém-criada. Em seguida, é criado um novo proxy que contém uma referência ao endereço do contrato do controlador recém-criado - somente o contrato do controlador pode invocar funções do proxy - uma restrição especificada no controlador e imposta pelo EVM. A Figura 8 fornece uma visão geral de uma interação entre um contrato inteligente e um aplicativo descentralizado no Ethereum.

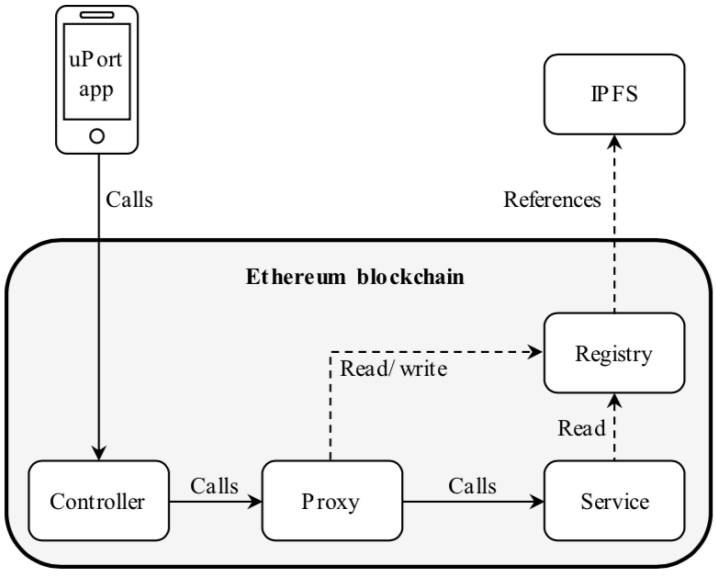


Figura 8. uPort elementos chave [38].

A chave privada que controla um uPortID é armazenada apenas no dispositivo móvel do usuário.

Portanto, um aspecto importante do uPort refere-se ao protocolo de recuperação de ID para o caso de perda ou roubo do dispositivo móvel do usuário. Para isso, os usuários devem indicar nominalmente uPortIDs que terão poder de voto que são os *trustees* (relação de confiança) no processo de substituição da chave pública do usuário solicitante. Um usuário é livre para criar vários uPortIDs que não são vinculados entre IDs.

Um aspecto final do esquema da uPort é o seu suporte para mapear com segurança os atributos de identidade para uma determinada uPortID. O registro da uPort é um contrato inteligente que armazena o mapeamento global de uPortIDs para atributos de identidade. Qualquer entidade pode consultar o registro, no entanto, apenas o proprietário de um uPortID específico pode modificar seus respectivos atributos.

Devido à ineficiência de armazenar grandes volumes de dados em um contrato inteligente, apenas o *hash* da estrutura do atributo JSON é armazenado no registro. Os dados em si são armazenados no IPFS: um sistema de arquivos distribuído em que um arquivo pode ser recuperado por seu *hash* criptográfico.

1.8.3.2 uPort Considerações Baseadas nas Leis da Identidade

A uPort não possui um servidor central e não autentica o proprietário de um uPortID, com isso, o risco de acesso não autorizado são transferidos para os métodos de autenticação local do dispositivo móvel do usuário. Embora o protocolo de recuperação social forneça um método para recuperar a propriedade de um uPortID perdido ou comprometido, os próprios *trustees* podem ser um vetor de ataque. A transparência fornecida a esses curadores, oferece a oportunidades de conluio contra um usuário específico da uPort.

Outro ponto é se um invasor puder comprometer um aplicativo da uPort e substituir os *trustes*, por meio do controlador, o uPortID ficará comprometido permanentemente. Assim, enquanto o uPort coloca mais controle sobre o uPortIDs nas mãos de seus usuários - uma vantagem para **(Lei 1)** - uma camada de complexidade e responsabilidade adicionais é inevitavelmente entregue aos usuários.

A uPort não requer divulgação de dados pessoais para iniciar um uPortID para um uso restrito e também respeita a privacidade em termos da falta de vinculação inerente entre os uPortIDs **(Lei 2)**. No entanto, o registro (quando usado) representa um ponto de centralização que pode ser investigado para obter informações sobre identificadores e dados de identidade. Portanto, embora os atributos específicos na estrutura de dados do atributo possam ser criptografados individualmente, a estrutura de dados JSON geral ainda é visível, o que poderia vazar metadados sobre atributos específicos ou relacionamentos com provedores de identidade/partes confiáveis/ *trustees*. Assim, há uma chance de que o excesso de confiança no registro possa comprometer a privacidade **(Lei 3)**.

Um aplicativo de comércio pode divulgar amplamente seu uPortID, mas o uPort não oferece nenhum diretório público para procurar os uPortIDs a partir de critérios de pesquisa arbitrários. A divulgação discreta de um uPortID é possível se um usuário criar novas uPortIDs para cada nova parte confiável que encontrar e evitar o uso do registro **(Lei 4)**. Embora um uPortID equivale a um contrato inteligente, um nó Ethereum honesto, mas curioso, poderia descobrir até mesmo URLs não divulgados através da análise do código de contrato inteligente armazenado em um determinado endereço para

determinar se é um modelo da uPort, isso implica em um grande trabalho e testes para descobrir se as uPortIDs não divulgadas são privadas na prática, mas com tempo e dinheiro tudo é possível.

A uPort não realiza nenhuma verificação de identidade, mas fornece uma estrutura para os usuários coletarem atributos de um ecossistema de provedores de confiança, mas simplesmente especifica o formato dos atributos armazenados em seu registro. Como consequência do proprietário da uPortID ter acesso de gravação à sua própria parte do registro, um usuário pode descartar seletivamente os atributos negativos que eles recebem, por exemplo: uma contagem de crédito negativa, por exemplo. **(Lei 5)**.

O aplicativo da uPortID fornece uma experiência de usuário consistente em todos os contextos de uso **(Lei 7)** devido à simples leitura de um código QR para iniciar interações com a outra parte confiável. Além disso, não é possível colocar representações de informações pessoalmente identificáveis na DLT o que prioriza a imutabilidade e a transparência dos dados **(Lei 6)**.

1.8.4 Identidade confiável descentralizada – ShoCard

ShoCard [41] fornece uma identidade confiável que aproveita o DLT para vincular um identificador de usuário, uma credencial confiável existente (por exemplo, passaporte, carteira de habilitação) e atributos de identidade adicionais, por meio de *hashes* criptográficos armazenados em transações Bitcoin. Os principais casos de uso da ShoCard são a verificação de identidade em interações presenciais e on-line.

1.8.4.1 uPort Abordagem

A ShoCard usa o plataforma Bitcoin como um serviço de registro de data e hora para *hashes* criptográficos e assinados das informações de identidade do usuário, que são extraídas do Bitcoin. O ShoCard incorpora um servidor central como parte essencial de seu esquema. Esse servidor é intermediário na troca de informações de identidade criptografada entre um usuário e a parte confiável. O esquema se baseia em três fases: (i) bootstrapping, (ii) certificação e (iii) validação. A Figura 9 apresenta essas fases.

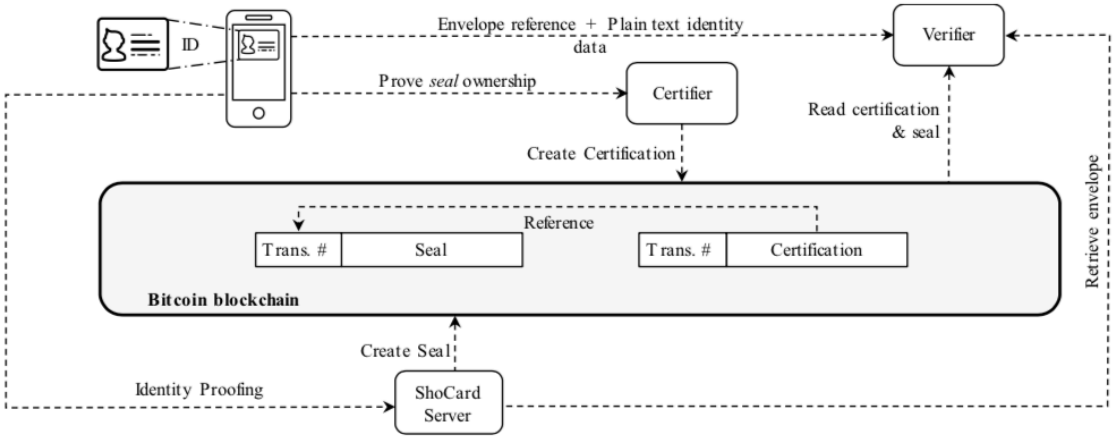


Figura 9. ShoCard elementos chave [38].

Bootstrapping ocorre na criação de um novo ShoCard. O aplicativo móvel ShoCard gera um novo par de chaves assimétricas para o usuário e escaneia suas credenciais de identidade usando a câmera do dispositivo. A varredura e os dados correspondentes são criptografados e armazenados no dispositivo móvel. O *hash*

assinado desses dados também é incorporado em uma transação Bitcoin para fins posteriores de validação de dados. O número de transação resultante do Bitcoin constitui o ShoCardID do usuário e é retido no aplicativo móvel como um ponteiro para o selo ShoCard.

Uma vez que um ShoCard é inicializado, o usuário pode interagir com provedores de identidade para reunir atributos adicionais em um processo chamado certificação. Para associar certificados a um ShoCardID, um provedor de identidade deve primeiro verificar se o usuário conhece os dados com *hash* para criá-lo e as chaves criptográficas que assinaram o selo. Em um contexto face-a-face, isso pode ser alcançado pelo usuário fornecendo os dados de identidade originais que formam o selo de seu dispositivo móvel, um desafio assinado digitalmente e apresentando a credencial confiável original. O certificado assume a forma de um *hash* assinado de novos atributos (e seu associado ShoCardID) em uma transação de Bitcoin criada pelo provedor. O provedor deve compartilhar o número da transação Bitcoin junto com um texto simples assinado dos novos atributos diretamente com o usuário. Como o usuário precisará posteriormente fornecer os atributos para as partes confiáveis e não deseja perdê-los se o dispositivo móvel for perdido, um servidor ShoCard oferece armazenamento para uma versão criptografada de certificações (conhecida como envelope). ShoCard nunca aprende a chave de criptografia, que permite ao usuário compartilhar certificações apenas com partes selecionadas.

A fase de validação ocorre quando uma parte confiante deve verificar uma certificação para determinar se um usuário tem o direito de acessar um serviço (por exemplo, fez o check-in em um voo). Para validar o envelope, o usuário deve primeiro fornecer à parte confiável a referência do envelope e sua chave de criptografia. Depois de recuperar o envelope dos servidores ShoCard, a parte confiante executa várias verificações: i) que a assinatura do envelope foi produzida com a mesma chave privada que assinou o selo; ii) que a assinatura da certificação foi criada por uma entidade de confiança e a certificação em texto simples corresponde àquela com *hash* e assinada na certificação; iii) finalmente, que os dados de identidade apresentados pelo usuário na transação pendente correspondem aos dados assinados e *hash* no selo.

1.8.4.2 ShoCard Considerações Baseadas nas Leis da Identidade

O servidor central ShoCard funciona como um intermediário para gerenciar a distribuição de certificações criptografadas entre os usuários do ShoCard e as partes confiáveis. Desta forma, o Sho-Card tem menos risco de violação de dados do que se armazenasse e distribuísse dados de identidade de texto simples. O armazenamento seguro de informações de identidade e o compartilhamento apropriado com terceiros confiáveis são controlados pelo usuário final **(Lei 1)**. No entanto, o papel intermediário de ShoCard cria incerteza sobre a existência longitudinal de um ShoCardID. Se a empresa deixasse de existir, os usuários do ShoCard não poderiam usar o sistema com as certificações adquiridas. Isso torna o ShoCard mais centralizado na prática do que sua confiança aberta no DLT poderia sugerir.

Cada identidade de ShoCard é inicializada com uma credencial confiável existente, como um passaporte ou carteira de motorista. Tal abordagem pode exigir que os usuários incorporem mais informações pessoais em seu selo ShoCard do que pretendiam originalmente. Isto pode tornar o ShoCard menos atraente para contas online de baixo valor **(Lei 2)**.

Como o usuário está no controle de iniciar atividades de compartilhamento e como o ShoCard armazena apenas dados criptografados, pode haver alguma confiança de que apenas partes justificáveis estão envolvidas na transação de compartilhamento de dados de identidade. No entanto, o servidor ShoCard pode ser capaz de associar um determinado ShoCardID a uma terceira parte confiável, já que os envelopes devem ser recuperados do servidor ShoCard pela terceira parte confiável **(Lei 3)**.

ShoCard suporta apenas identificadores unidirecionais e não possui o conceito de registro público de ShoCardIDs. Embora identificadores omnidirecionais possam ser necessários no futuro para realizar sua visão de um ecossistema de certificações reutilizáveis **(Lei 4)**.

A ShoCard suporta uma infinidade de provedores de identidades diferentes por meio de sua funcionalidade de certificação, mas esses provedores devem criar uma integração personalizada com os próprios serviços web da Sho-Card, além do Bitcoin, o que poderia ser uma barreira à aceitação. A decisão de realizar essa integração pode ser motivada pela confiabilidade da comprovação de identidade de seus usuários pelo Sho-Card **(Lei 5)**.

A digitalização de documentos de identidade e baseado em códigos QR o que é um paradigma de interação dominante na experiência do usuário do ShoCard - é simples e consistente **(Lei 7)**. No entanto, não está claro quais seriam as motivações do usuário para adotar esse novo tipo de identidade digital, e como os usuários seriam educados sobre as implicações da referência de dados de identidade em rede *blockchain* **(Lei 6)**. Os usuários também não são compatíveis com o gerenciamento de chaves criptográficas.

Um último ponto diz respeito à disponibilidade geral do ShoCard. As transações de Bitcoin levam em média 10 minutos para serem exploradas na rede *blockchain* e, além disso, é recomendável aguardar que seis blocos adicionais sejam extraídos antes de assumir a liquidação de uma transação. Isso poderia levar em média o tempo de espera para a liquidação a uma hora. Se um contexto depender da liquidação em tempo real das certificações, essa prazo poderá criar desafios com relação a experiência do usuário e a adoção pelos fornecedores.

1.9 Aspectos legais e Padronização da Identidade Digital Autossoberana

a) Aspectos Legais

Um dos grandes desafios da identidade digitais autossoberana é o atendimento dos requisitos das principais leis relacionadas com a proteção de dados pessoais. Vários países adotaram um modelo jurídico para proteção de dados pessoais através de um regime legal de proteção de dados, na forma de uma lei geral. Com exceção dos Estados Unidos, a maioria dos países desenvolvidos, e também o Brasil, aprovaram leis abrangentes contemplando os setores públicos e privado. Embora alguns países possuam suas leis gerais, estas podem coexistir com normas setoriais, regulando setores específicos de forma complementar às leis gerais [42].

Dentre estas leis, destaca-se o Regulamento Geral de Proteção de Dados da União Europeia (RGPD), também conhecido como *General Data Protection Regulation* (GDPR), que foi elaborado pelo Parlamento Europeu e Conselho da União Europeia e publicado no dia 04 de maio de 2016. Ele foi implementado nos 28 países membros da União Europeia em 25 de maio de 2018. Ele se aplica à proteção das pessoas naturais no que diz respeito à proteção de dados e também ao livre movimento desses dados e

revoga a Diretiva de Proteção de Dados Pessoais de 1995 (95/46/CE). O regulamento, que na União Europeia tem força de lei, possui um conteúdo bastante extenso, com 173 considerandas e 99 artigos.

No Brasil, a Lei 13.709/2018, também conhecida como Lei Geral de Proteção de Dados brasileira (LGPD), que foi publicada em 14 de agosto de 2018 e, segundo COTS [43], com esta publicação "o Brasil se integrou, não sem um certo atraso, ao grupo de países que possuem legislações específica para proteção de dados pessoais". Pode-se afirmar que a grande fonte de inspiração para a elaboração da LGPD foi o GDPR, sendo a primeira mais genérica e, consequentemente, menos detalhada que o regulamento.

Seguem algumas questões relevantes das leis gerais de proteção de dados pessoais que podem impactar soluções que utilizam blockchain, com destaque para a identidade digital autossoberana:

- **Direto de apagar ou direito de ser esquecido**: Em algumas algumas situações, o titular tem o direito de obter do responsável pelo tratamento o apagamento dos seus dados pessoais, sem demora injustificada, e este tem a obrigação de apagar os dados pessoais, sem demora injustificada. Portanto, tal direito inviabiliza o registro de dados pessoais na ledger. Nas propostas de identidade autossoberana, os dados pessoais nunca são colocados na ledger. Em vez disso, são colocados somente identificadores pseudônimos e descentralizados denominados *Decentralized Identifiers* (DIDs) [44], chaves públicas pseudônimas, endereços de agentes e as estruturas das credenciais emitidas (schemas), conforme especificado pela W3C [45] . Isso permite que a troca de dados pessoais ocorra inteiramente fora da ledger. Vale destacar que, diferentemente do GDPR, na LGDP não tem previsão específica para tratamento do direito de ser esquecido. Segundo ministro do STJ Paulo de Tarso Sanseverino "A LGPD abrange todos os dados pessoais, inclusive digitais. O Marco Civil tem a preocupação somente com os efeitos da Internet. Apesar disso, a nova legislação não tem previsões importantes, como é o caso do Direito ao esquecimento" [46]
- **Direito de retificação**: O titular tem o direito de obter, sem demora injustificada, do responsável pelo tratamento a retificação dos dados pessoais inexatos que lhe digam respeito. Como os dados pessoais não são colocados na *ledger* e, geralmente, ficam sobre a gestão do titular, este requisito também é atendido pelas soluções de identidade digital autossoberana;
- **Direito de acesso**: Isso significa que os titulares de dados têm o direito de perguntar a um controlador de dados se seus dados pessoais estão sendo processados e, se forem, receber detalhes sobre como este processamento se dá e onde. No caso da identidade digital autossoberana quem controla o acesso aos dados é o próprio titular através dos DIDs;
- **Portabilidade dos dados**: O direito à portabilidade de dados (artigo 20.º do GDPR, por exemplo) permite que um titular de dados receba dados de um responsável pelo tratamento, a fim de os transmitir a outro controlador [47]. O Grupo de Trabalho do Artigo 29[1], por exemplo, considera que o "principal

[1] Trata-se da designação abreviada do Grupo de Proteção de Dados estabelecido pelo artigo 29.º da Diretiva 95/46/CE. Proporciona à Comissão Europeia aconselhamento independente sobre questões de proteção de dados e contribui para o desenvolvimento de políticas harmonizadas de proteção de dados nos Estados-Membros da UE. O grupo de trabalho é composto por todos os representantes das autoridades nacionais de supervisão dos Estados-Membros da UE.

objetivo da portabilidade de dados é aumentar o controle dos indivíduos sobre seus dados pessoais e garantir que eles desempenhem um papel ativo no ecossistema de dados". A maioria das soluções atuais não fornecem aos proprietários tal funcionalidade. Isso não se aplica a identidade digital autossoberana, onde a gestão dos dados (armazenamento e controle de acesso) é definida pelo próprio usuário dos dados, podemos armazenar estes dados nos seus próprios dispositivos através de um *Mobile Edge Agent*[2] ou ainda na nuvem usando um *Hub* que armazena e compartilha dados em nome dos seus proprietários, podendo ser concebido como uma carteira remota, que armazena todos os dados anonimamente criptografados, mas não as chaves.

b) Os esforços de padronização

Atualmente, existem ações globais no sentido de buscar padronização para os diferentes protocolos e agentes que constituem as soluções de identidade digital autossoberana. Conforme apresentado na figura X [48], existem vários órgãos envolvidos nestes esforços de padronização, com destaque para o W3C[3] (*World Wide Web Consortium*) envolvido na padronização dos DIDs e das *Verifiables Credentials*.

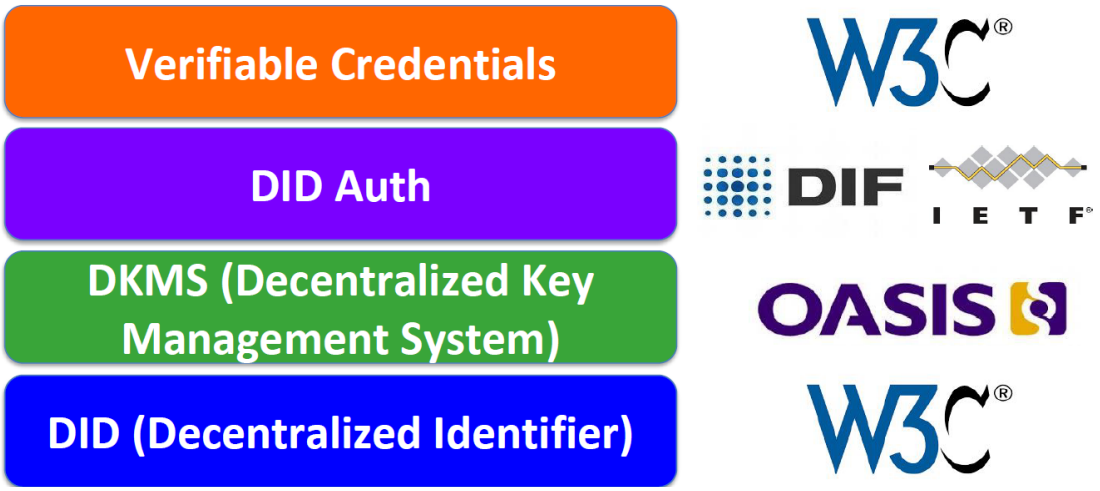


Figura 10 - Esforços de Padronização

Muitos destes padrões ainda estão em fase de discussão. Um exemplo é a própria padronização dos DIDs, com o W3C lançando recentemente a versão V.013 do *Data Model and Syntaxes* [49].

DKMS (*Decentralized Key Management System* ou Sistema de Gerenciamento de Chaves Descentralizadas) é um padrão aberto emergente para gerenciar DIDs e chaves privadas. O DKMS se aplica às carteiras onde se armazenam DIDs e chaves privadas, assim como aos agentes que leem/escrevem nessas carteiras. A ideia do DKMS é padronizar as carteiras para que o usuário nunca precise se preocupar com a segurança, a privacidade ou o bloqueio de fornecedores. A arquitetura inicial do DKMS está agora em análise pública aberta no *Hyperledger Indy github* [50].

[2] Mobile Agents são aplicações com as quais as pessoas interagem diretamente para controlar sua identidade, gerenciando convites para criar Identificadores Descentralizados (DIDs) com outros agentes, gerenciando a concessão de dados pessoais e etc. Tudo isso no dispositivo mobile do dono da identidade.

[3] W3C - O World Wide Web Consortium é a principal organização de padronização da World Wide Web.

O *DID Auth* é uma forma padrão simples para um proprietário de DID autenticar, comprovando o controle de uma chave privada. Em novembro de 2017 foi formado o grupo de trabalho *DID Auth* pela *Decentralized Identity Foundation*[4]. Em fevereiro de 2018, foram lançadas as especificações e implementações preliminares do padrão. Em abril de 2018, foi apresentado o primeiro protótipo do padrão no *Internet Identity Workshop*.

As *Verifiable Credentials* é um formato para credenciais digitais interoperáveis e criptograficamente verificáveis que estão sendo definidas pelo *Verifiable Claims Working Group* do W3C criado em maio de 2017. A missão do grupo é tornar mais fácil e mais segura a divulgação e troca de credenciais que foram verificadas por terceiros [51].

Por enquanto, a discussão sobre padronização de identidade digital autossoberana no Brasil está sendo contemplada na comissão ABNT/CEE-307 - *Blockchain* e Tecnologias de Registro Distribuídas, que possui no seu âmbito de atuação a normalização no campo de *Blockchain* e tecnologias de registro distribuídas, no que concerne a terminologia e generalidades. Esta Comissão é espelho do ISO/TC-307 – *Blockchain* and *Distributed Ledger Technologies.*

Recentemente, a comissão divulgou a versão preliminar do documento "*Blockchain* e tecnologias de registro distribuídas" – Conceitos e elementos da tecnologia *Blockchain*" composto por seis partes. O tema identidade está sendo abordado na sexta parte do documento (segurança, privacidade e identidade).

1.10 A importância e os desafios das Carteiras Digitais (*"Digital Wallets"*)

Carteiras Digitais são peça chave para soluções de Identidade Digital Autossoberana, elas não servem apenas para guardarmos de forma segura e confiável nossas credenciais, chaves privadas e informações, além disso, elas servem também para gerenciarmos para que agentes terceiros possam, com o nosso consentimento, acessar às nossas credenciais e dados, nos dando meios para verificar quem possui tais permissões e também poder para revogar permissões que não queiramos mais ou que não fazem mais sentido. Além disso, Carteiras Digitais são responsáveis por fazer o gerenciamento de nossas credenciais, desde comprovantes de conclusão de cursos, que devem ficar por um longo tempo (pelo menos enquanto forem relevantes) até mesmo a simples ingressos para eventos (por exemplo, cinema e teatro), que poderiam ser descartados (ou movidos para um arquivo morto) apenas a algumas horas depois de seu uso.

Apesar de indubitável que carteiras digitais será amplamente usada para o gerenciamento e controle de nossas identidade e dados, seu conceito possui mais de duas décadas. Um exemplo de uma carteira digital criada no início dos anos dois mil, é o Microsoft Passport [52], que se propunha a ser a solução informações para *"single signon"* e de cartões de créditos em um único lugar. O Microsoft Passport fracassou, sem nunca ter chegado perto da quantidade de usuários e aplicações integradas que um dia se pensou para ele. Assim como o Microsoft Passport, inúmeras outras soluções de Carteiras Digitais fracassaram ao longo dos últimos vinte anos, em geral, os principais motivos de fracassos foram, dentre outros, os seguintes aspectos:

[4] O DIF é uma organização focada no desenvolvimento dos elementos fundamentais necessários para estabelecer um ecossistema aberto para a identidade descentralizada e garantir a interoperabilidade entre todos os participantes.

- As Carteiras Digitais criadas nesse período, em sua maioria, são iniciativas fechadas, não possuem padronização e ou meios para portabilidade, fazendo com que os usuários se tornassem reféns dessas soluções;
- Problemas de segurança, como os reportados para a abordagem de *single signon* [53];
- Essas iniciativas foram ambiciosas demais, propondo-se a gerenciar e a controlar todas as credenciais dos usuários de uma única vez, inevitavelmente se tornando, por tanto, soluções complexas demais e, em geral, enfadonhas de se usar.

Além dos desafios citados acima, que continuam sendo uma realidade a ser enfrentada por Carteiras Digitais modernas, também vale ressaltar outros desafios como, por exemplo [54]:

- Portabilidade entre carteiras (o usuário, tendo controle de sua identidade, deve ser capaz e ter meio para portar e controlar suas credenciais na solução que preferir e mais confiar);
- Padronização de credenciais, sendo que, em tese, o *schema* de uma credencial financeira emitida por um *agent*, deveria ser minimamente compatível com o esquema da credencial de um outro *agent* qualquer do mesmo segmento;
- Facilidade para realizar backup, vez que acidentes acontecem e, perda de dispositivos móveis (onde, eventualmente, as carteiras digitais de pessoas físicas poderiam ficar armazenadas), ou até mesmo em caso de dano de aparelho ou exposição da carteira, em todos esses casos, deveria ser possível ao usuário revogar suas credenciais expostas e conseguir emitir novas e recuperar os dados perdidos;
- E por último, mas tão crucial quanto os demais, é a usabilidade das carteiras digitais. Carteiras digitais confusas, com enormes quantidades de credenciais antigas que não fazem mais sentido e que não são fáceis de usar ou cujas funções não agreguem valor e ou facilidade ao seu portador, são os grandes motivos para que usuários não façam o uso ordinário das novas carteiras digitais, como por exemplo, Google Wallet ou Apple Wallet.

Além dos desafios acima demonstrados, será também necessária a criação de metodologias e, eventualmente, até mesmo de certificação de segurança para carteiras digitais, afins de garantir ante ao portador da credencial que aquela solução que armazenará sua representação digital ante ao mundo (isto é, sua identidade digital) é confiável e segura o bastante para que ele possa usá-la

1.11 Criptografias BLS-Signature e CL-Signature para criação de verfinym e pseudonym

O framework para gerenciamento de Identidade Digital Descentralizada do projeto Hyperledger, o Hyperledger Indy, possui uma biblioteca de criptografia chamada *"indy-crypto"*, que atualmente é usado para "Provas Zero de Conhecimento" (*"ZKPs"*) e para esquemas de assinaturas digitais para credenciais verificáveis (*"Verifiable Credentials"*) emitidas entre os agentes.

Resumidamente, um esquema de assinatura digital possui ao menos as três principais funções a seguir:

- **Geração de um par de chaves**, pública e privada (*"key pair"*);
- Processo de operação para **assinatura digital de uma mensagem** que, em geral (mas, não exclusivamente), é realizado com o uso da chave privada;
- **Verificação da assinatura digital**, usando a chave pública.

O esquema de assinatura adotado pelo framework Hyperledger Indy para permitir verificar se a assinatura é verdadeira, é o Boneh-Lynn-Shacham (BLS) [55].

O esquema BLS usa o conceito de emparelhamento bilinear (*"bilinear paring"*) [56] para as operações de verificações de assinaturas, fazendo o gerenciamento das assinaturas por grupos de curvas elípticas (cada assinatura pertence a um determinado grupo de curva elíptica), tal abordagem de trabalhar com grupos de curvas elípticas provê defesas sólidas e eficientes contra ataques de *Index Calculus* [57], permitindo assim a criação de assinaturas menores que as do padrão FDH, mas com o mesmo nível de segurança [58].

O emprego do esquema BLS permite também, a agregação de múltiplas assinaturas para uma mesma mensagem, criando assim, a possibilidade de realizar uma análise criptográfica a fim de constatar qual o remetente(s) e o destinatário de uma determinada mensagem assinada [59] com esquema BLS, sendo esta, então, a abordagem do conceito de credenciais verificáveis (*"Verifiable Credentials"* e *"Verifinym"*) do Hyperledger Indy.

Além da BLS, outro esquema de assinatura digital encapsulado pela biblioteca indy-crypto é o Camenisch-Lysyanskaya (CL), esquema este usado para a geração de validações ZKPs e a criação de credenciais anonimizadas (*"Pseudonym"*) [60].

Esses dois esquemas de assinaturas digitais, mais o protocolo *Blockchain* específico para o gerenciamento de transações para Identidade Digital Autossoberana, são os pilares do framework Hyperledger-Indy e o surgimento de uma rede para Identidade Digita Autossoberana.

1.12 Requisitos de sistema

Para o correto desenvolvimento de soluções para Hyperledger Indy, o sistema operacional que deve ser utilizado é o Ubuntu 16.04 LTS com a seguinte configuração mínima de hardware:

- Processador Core-i5 terceira geração ou superior;
- 6 Gigabytes de memória RAM ou superior;
- Mínimo de 80 Gigabytes de memória em disco disponível.

1.13 Tutorial Hyperledger Indy

Para preparar o ambiente de desenvolvimento voltado para .Net são necessários alguns pré-requisitos.

Visual Studio Code.

https://code.visualstudio.com/

- Após a sua instalação adicione as extensões C# e *Nuget package* manager

.Net Core na versão 2.2.

Instalando o Indy-SDK.

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
68DB5E88
sudo add-apt-repository "deb https://repo.sovrin.org/sdk/deb
xenial stable"

sudo apt-get update
sudo apt-get install -y libindy
```

Se você baixar o repositório libindy da *master* verifique a biblioteca libindy.so está em /usr/lib/libindy.so senão copie para este diretório.

Se preferir há a opção construir o ambiente de desenvolvimento gerando os binários a partir do código fonte. Siga os passos descritos no link [https://github.com/hyperledger/indy-sdk/blob/master/docs/build-guides/ubuntu-build.md]
Instalando o *Docker*:
https://docs.docker.com/

A maneira recomendada de iniciar um nó de rede local (um pool) é utilizando o *Docker*. Execute os seguintes comandos no terminal. Em seguida configure o IP do *Docker* container e do pool de forma que eles correspondam.

```
docker build -f ci/indy-pool.dockerfile -t indy_pool .
docker run -itd -p 9701-9708:9701-9708 indy_pool
```

1.14 Criando um projeto.

Com o ambiente já configurado, iniciaremos o tutorial com operações básicas da biblioteca libindy. Em nosso exemplo vamos criar uma negociação de prova de confiança. Para elucidar, imagine que uma instituição peça informações para você e, para provar, você apresente um documento emitido pelo governo ou órgão de confiança sob a governança da uma federação.

A negociação de prova normalmente começa quando um verificador solicitar uma prova de veracidade de informação. Uma solicitação de prova é um arquivo JSON que descreve que tipo de prova satisfaria a parte confiável. Depois que a solicitação de prova é recebida, um detentor de credenciais deve verificar sua carteira de identidade para descobrir quais credenciais podem ser usadas para atender à solicitação

Para iniciar, crie um projeto no *Visual Code* (neste exemplo um projeto do tipo console *application*):

```
dotnet new console -lang C# -n <nome-do-seu-projeto>
```

Importe o pacote *HyperLedger Indy SDK* e *Newtonsoft* via *nuget* para o projeto
https://www.nuget.org/packages/Hyperledger.Indy.Sdk
https://www.nuget.org/packages/Newtonsoft.Json/

Crie um arquivo na raiz do projeto com o seguinte conteúdo abaixo e prossiga da mesma forma para criar um total de 4 nós de rede ou copie docker_pool_transactions_genesis em "indy-sdk/cli/docker_pool_transactions_genesis" e para mapear o container docker mude o "client_ip" e o "node_ip" para de "10.0.0.2" para "127.0.0.1" para usar localhost.

Configuração do nó 1:

```
{
   "reqSignature": {},
   "txn": {
       "data": {
           "data": {
               "alias": "Node1",
               "blskey":
"4N8aUNHSgjQVgkpm8nhNEfDf6txHznoYREg9kirmJrkivgL4oSEimFF6
nsQ6M41QvhM2Z33nves5vfSn9n1UwNFJBYtWVnHYMATn76vLuL3zU88Ky
eAYcHfsih3He6UHcXDxcaecHVz6jhCYz1P2UZn2bDVruL5wXpehgBfBaL
Km3Ba",
               "blskey_pop":
"RahHYiCvoNCtPTrVtP7nMC5eTYrsUA8WjXbdhNc8debh1agE9bGiJxWB
XYNFbnJXoXhWFMvyqhqhRoq737YQemH5ik9oL7R4NTTCz2LEZhkgLJzB3
QRQqJyBNyv7acbdHrAT8nQ9UkLbaVL9NBpnWXBTw4LEMePaSHEw66RzPN
dAX1",
               "client_ip": "127.0.0.1",
               "client_port": 9702,
               "node_ip": "127.0.0.1",
               "node_port": 9701,
               "services": [
                   "VALIDATOR"
               ]
           },
           "dest":
"Gw6pDLhcBcoQesN72qfotTgFa7cbuqZpkX3Xo6pLhPhv"
       },
       "metadata": {
           "from": "Th7MpTaRZVRYnPiabds81Y"
       },
       "type": "0"
   },
   "txnMetadata": {
       "seqNo": 1,
```



```
       "txnId":
"fea82e10e894419fe2bea7d96296a6d46f50f93f9eeda954ec461b2e
d2950b62"
   },
   "ver": "1"
}
```

Copie as seguintes bibliotecas:

```
using System;
using Hyperledger.Indy.PoolApi;
using Hyperledger.Indy.WalletApi;
using Hyperledger.Indy.LedgerApi;
using Hyperledger.Indy.DidApi;
using Hyperledger.Indy.AnonCredsApi;
using Newtonsoft.Json.Linq;
```

Passo 1: Criando um DID

Nesta primeira parte será criado um identificador descentralizado (DID) e posteriormente será realizada a consulta de informação na *ledger* em função do *Trust Anchor* (nó de confiança). Para isso, será necessário um *pool* de nós do Hyperledger Indy para as transações na *ledger*. Na sequência, será criada uma carteira digital onde para controle manipular uma carteira onde será alocado o DID com a par de chaves e as credenciais do *Trust Anchor* e do Steward. Posteriormente será feito uma requisição para o Steward submeter o DID do *Trust Anchor* (nó de confiança) na *ledger* com a assinatura de ambos com as suas respectivas chave privada.

A título de exemplo usaremos apenas uma carteira, mas os agentes e Steward devem ter suas próprias carteiras para estabelecer uma canal de comunicação em uma situação real.

Na configuração do pool é necessário referenciar o diretório do docker pool. Lembrando que o endereço IP do container docker deve estar devidamente configurado e corresponder ao do pool. Certifique-se de subir o container criado e copie o código abaixo para a classe demo do projeto

Obs: O *seed* do did declarado inicialmente deve conter 32 caracteres e funciona como uma chave de administrador da rede.

```
Console.WriteLine("Selecionando a versão Indy Node 2.0");
       Pool.SetProtocolVersionAsync(2).Wait();

       string poolName = "pooldemo";
       string startupPath = Environment.CurrentDirectory;
       string poolConfig = "{\"genesis_txn\": \"" +
startupPath + "/docker_pool_transactions_genesis\"}";
       string issuerWalletConfig =
"{\"id\":\"issuerwallet\"}";
       string issuerWalletCredencial =
"{\"key\":\"issuerwalletkey\"}";
       string did = "{\"seed\":
\"000000000000000000000000Steward1\"}";
```

```
      Console.WriteLine("# 1  Criando a configuração de
um pool local na ledger que será usado posteriormente
para conectar o pool de nós da rede.");
      Pool.CreatePoolLedgerConfigAsync(poolName,
poolConfig).Wait();

      Console.WriteLine("# 2 Abrindo o pool na ledger e
obtendo seu gerenciador da libindy.");
      Pool poolHandle =
Pool.OpenPoolLedgerAsync(poolName, "{}").Result;

      Console.WriteLine("# 3 Criando uma  identity
wallet.");
      Wallet.CreateWalletAsync(issuerWalletConfig,
issuerWalletCredencial).Wait();

      Console.WriteLine("# 4 Abrindo identity wallet e
obtendo seu gerenciador da libindy.");
      Wallet issueWalletHandle =
Wallet.OpenWalletAsync(issuerWalletConfig,
issuerWalletCredencial).Result;

      Console.WriteLine("# 5 Gerando e armazenando o DID
Steward e a chave-valor(verkey).");
      CreateAndStoreMyDidResult DidSteward =
Did.CreateAndStoreMyDidAsync(issueWalletHandle,
did).Result;
      Console.WriteLine(string.Format("Steward Did:
{0}", DidSteward.Did));
      Console.WriteLine(string.Format("Steward
Verkey:{0}", DidSteward.VerKey));

      Console.WriteLine("# 6 Gerando e armazenado o Did
do Trust Anchor e a verkey");
      CreateAndStoreMyDidResult DidTrustAnchor =
Did.CreateAndStoreMyDidAsync(issueWalletHandle,
"{}").Result;
      Console.WriteLine(string.Format("Trust Anchor DID:
{0}", DidTrustAnchor.Did));
      Console.WriteLine(string.Format("Trust Anchor
Verkey: {0}", DidTrustAnchor.VerKey));

      Console.WriteLine("#7 Construindo uma requisição
Nym para adicionar o Trust Anchor na ledger");
      var nymRequest =
Ledger.BuildNymRequestAsync(DidSteward.Did,
DidTrustAnchor.Did, DidTrustAnchor.VerKey, null,
"TRUST_ANCHOR").Result;
```



```
       Console.WriteLine(string.Format("Nym Request:
{0}", nymRequest));

       Console.WriteLine("#8 Enviando a requisição Nym
para a ledger");
       var nymResponse =
Ledger.SignAndSubmitRequestAsync(poolHandle,
issueWalletHandle, DidSteward.Did, nymRequest).Result;
       Console.WriteLine(string.Format("Nym Response:
{0}", nymResponse));
```

Passo 2: Gerando uma Credencial

Nesta segunda parte da demonstração, veremos como um emissor de credenciais cria e define um *schema*, que é um documento *json* com um conjunto de atributos específicos que irão fazer parte de uma credencial. O Steward fará, portanto, uma requisição para adicionar o *schema* na *ledger*.

Criaremos então, uma credencial que faz referência ao *schema* criado e define quem irá emitir credenciais com ele e que tipo de método de assinatura é utilizado entre outras informações. Após esse processo de definição, o *Trust Anchor*, que será o emissor de credenciais no caso, usa a credencial anônima para armazená-la (*Claim Credencial*), utilizando o método de assinatura CL (Camenisch Lysychansk) de Zero *Knowledge Prove*.

```
Console.WriteLine("#9 O Emissor cria um Schema de
credencial com informações que serão requisitadas");
       String name = "gvt";
       String version = "1.1";
       String attributes =
"[\"age\",\"sex\",\"height\",\"name\"]";
       IssuerCreateSchemaResult issuerCreateSchema =
AnonCreds.IssuerCreateSchemaAsync(DidSteward.Did, name,
version, attributes).Result;
       Console.WriteLine("Schema : {0}",
issuerCreateSchema);

       Console.WriteLine("#10 Criar uma requisição para
adicionar um novo Schema na ledger");
       var schemaRequest =
Ledger.BuildSchemaRequestAsync(DidSteward.Did,
issuerCreateSchema.SchemaJson).Result;
       Console.WriteLine("Schema Request {0}",
issuerCreateSchema.SchemaJson);

       Console.WriteLine("#11 Enviar a requisição do
Schema para a ledger e obtém o resultado");
```

```
       var schemaResponse =
Ledger.SignAndSubmitRequestAsync(poolHandle,
issueWalletHandle, DidSteward.Did, schemaRequest).Result;
       Console.WriteLine("Schema Response: {0}",
schemaResponse);

       Console.WriteLine("#12 Criando e armazenando a
definição da credencial para o Schema entregue em função
do Trust Anchor");
       string configJson =
"{\"support_revocation\":false}";
       string tag = "TAG1";
       var credDef =
AnonCreds.IssuerCreateAndStoreCredentialDefAsync(issueWal
letHandle, DidTrustAnchor.Did,
issuerCreateSchema.SchemaJson, tag, null,
configJson).Result;

       Console.WriteLine("ID da Credencial:\n" +
credDef.CredDefId);
       Console.WriteLine("Definição da Credencial
JSON:\n" + credDef.CredDefJson);
```


Passo 3: Verificando uma Credencial

Nessa sessão será criado uma negociação de prova de confiança onde será verificado junto ao nó de confiança se a informação requisitada por um verificador é verdadeira com base no *schema* e na credencial gerada. Para isso, vamos gerar uma carteira para operar transações de provas de informações. Para provar que a credencial realmente pertence ao detentor, será inserido um pedaço de informação oculto chamado *link secret* na credencial.

```
Console.WriteLine("#13 Criando uma carteira para operar as
transações de prova de informação");
       string proverDid = "VsKV7grR1BUE29mG2Fm2kX";
       string proverWalletConfig =
"{\"id\":\"prover_wallet\"}";
       string proverWalletCredential =
"{\"key\":\"prover_wallet_key\"}";

       Wallet.CreateWalletAsync(proverWalletConfig,
proverWalletCredential).Wait();
       Wallet proverWalletHandle =
Wallet.OpenWalletAsync(proverWalletConfig,
proverWalletCredential).Result;

       Console.WriteLine("#14. Criação do Link Secret");
       string proverLinkSecretName = "link Secret";
       var linkSecretId =
AnonCreds.ProverCreateMasterSecretAsync(proverWalletHandle
, proverLinkSecretName).Result;

       Console.WriteLine("#15 Emissor (Trust Anchor) cria
uma oferta para o receptor da credencial");
       var credOffer =
AnonCreds.IssuerCreateCredentialOfferAsync(issueWalletHand
le, credDef.CredDefId).Result;
       Console.WriteLine("Credencial Offer : {0}",
credOffer);

       Console.WriteLine("#17 O link secret é enviado
junto a credencial requisitada");
       var credRequest =
AnonCreds.ProverCreateCredentialReqAsync(proverWalletHandl
e, proverDid, credOffer, credDef.CredDefJson,
proverLinkSecretName).Result;
       Console.WriteLine("Credencial Request: {0}",
credRequest);

       Console.WriteLine("#18 Emissor (Trust Anchor) cria
a credencial requisitada baseada no Schema");
       string credValuesJson = "{\"sex\": {\"raw\":
\"male\", \"encoded\":
```

```
\"594465709955896723921094925839488742869205008160769251991705001114423311510
3\" },"
       + "\"name\": { \"raw\": \"Alex\", \"encoded\":
\"992628570980577103383069676095884100256486223083942506668496655324486122028
74\" },"
       + "\"height\": { \"raw\": \"175\", \"encoded\":
\"175\" },"
       + "\"age\": { \"raw\": \"28\", \"encoded\": \"28\"
}}";

       var credJson =
AnonCreds.IssuerCreateCredentialAsync(issueWalletHandle,
credOffer, credRequest.CredentialRequestJson,
credValuesJson, null, null).Result;
       Console.WriteLine("Credencial Json : {0}",
credJson);

       Console.WriteLine("#19 A credencial de prova
recebida é processada e armazenada");
       var proverCredencial =
AnonCreds.ProverStoreCredentialAsync(proverWalletHandle,
null, credRequest.CredentialRequestMetadataJson,
credJson.CredentialJson, credDef.CredDefJson,
null).Result;
```

A partir de então, o emissor e a prova que recebe a credencial estabelecem uma relacionamento interativo. O emissor oferece uma credencial a um agente, que faz a requisição desta credencial enviando de forma oculta o *link secret*. Então o emissor fornece a credencial que agora estará na posse do agente em questão.

Passo 4: Autenticando Credenciais

Neste ponto, o titular gera e apresenta uma prova. Isso é feito através da construção de um JSON que seleciona as credenciais que para satisfazer a solicitação de prova com os devidos atributos. A prova é criada com a função proverCreateProof com os parâmetros apropriados. Após a validação dos atributos finalizaremos nosso exemplo efetuando fechamento e deleção das carteiras e o pool criados.

```
Console.WriteLine("#20 Obtendo a requisição de prova\n");
       string proofRequestJson = "{"
           + "\"nonce\": \"123432421212\","
           + "\"name\": \"proof_req_1\","
           + "\"version\": \"0.1\","
           + "\"requested_attributes\": {"
```



```
            + "\"attr1_referent\": {"
            + "\"name\": \"name\","
            + "\"restrictions\": [{"
            + "\"cred_def_id\": \"" + credDef.CredDefId +
"\"}]}},"
            + "\"requested_predicates\": {"
            + "\"predicate1_referent\": {"
            + "\"name\": \"age\","
            + "\"p_type\": \">=\","
            + "\"p_value\": 18,"
            + "\"restrictions\": [{"
            + "\"issuer_did\": \"" + DidTrustAnchor.Did +
"\""
            + "}]}}}";

        Console.WriteLine("Request da prova: {0}",
proofRequestJson);

        Console.WriteLine("# 21 Fazendo uma busca na
credencial pelo atributo a ser verificado(como idade por
exemplo)");
        var proverCredSearch =
AnonCreds.ProverSearchCredentialsForProofRequestAsync(pro
verWalletHandle, proofRequestJson).Result;

        string credForAttr1String =
proverCredSearch.NextAsync("attr1_referent", 1).Result;
        string referent =
JArray.Parse(credForAttr1String).First["cred_info"].Value
<string>("referent");
        Console.WriteLine(string.Format("Credencial de
prova do attr1:{0}", credForAttr1String));

        Console.WriteLine("Montando uma requisição de
prova");
        string requestedCredentialsJson = "{"
           + "\"self_attested_attributes\": {},"
           + "\"requested_attributes\": {"
           + "\"attr1_referent\": {"
           + "\"cred_id\": \"" + referent + "\","
           + "\"revealed\": true"
           + " }},"
           + "\"requested_predicates\":{"
           + "\"predicate1_referent\":{"
           + "\"cred_id\": \"" + referent + "\""
           + "}}}";

        Console.WriteLine(string.Format("Credencial
requerida para prova:{0}", requestedCredentialsJson));
```

```
       string schemasJson = new JObject(new
JProperty(issuerCreateSchema.SchemaId,
JObject.Parse(issuerCreateSchema.SchemaJson))).ToString()
;
       string credentialDefsJson = new JObject(new
JProperty(credDef.CredDefId,
JObject.Parse(credDef.CredDefJson))).ToString();

       var proofJson =
AnonCreds.ProverCreateProofAsync(proverWalletHandle,
proofRequestJson, requestedCredentialsJson, linkSecretId,
schemasJson, credentialDefsJson, "{}").Result;
       string revRegDefsJson = "{}";
       string revRegsJson = "{}";

       Console.WriteLine("Verificando se as informações
são verdadeiras");
       var valid =
AnonCreds.VerifierVerifyProofAsync(proofRequestJson,
proofJson, schemasJson, credentialDefsJson,
revRegDefsJson, revRegsJson).Result;
       Console.WriteLine("Prova :");
       Console.WriteLine("Nome= {0}",
JObject.Parse(proofJson)["requested_proof"]["revealed_att
rs"]["attr1_referent"].Value<string>("raw"));
       Console.WriteLine("Verificado= {0}", valid);
 Console.WriteLine("Fechando as carteiras");
       issueWalletHandle.CloseAsync().Wait();
       proverWalletHandle.CloseAsync().Wait();

       Console.WriteLine("Deletando as carteiras
criadas");
       Wallet.DeleteWalletAsync(issuerWalletConfig,
issuerWalletCredencial).Wait();
       Wallet.DeleteWalletAsync(proverWalletConfig,
proverWalletCredential).Wait();

       Console.WriteLine("Deletando o pool ledger");
       poolHandle.CloseAsync().Wait();
       Pool.DeletePoolLedgerConfigAsync(poolName).Wait();

   }
```


Referencias

[1] Histórias e Insumos. 'Revolução do Período Neolítico. Disponível: http://www.historiaresumos.com/revolucao-periodo-neolitica. Acessado Junho de 2019.

[2] Brealey, R., Myers, S., Allen, F. 'Princípios de Finanças Corporativas'. McGraw-Hill. 2013.

[3] IBM. '*Blockchain sharing economy*'. Disponível: https://www.ibm.com/developerworks/br/library/iot-Blockchain-sharing-economy/index.html. Acessado Junho de 2019.

[4] Tapscott, D.; Tapscott, A. '*Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World*'. Maio de 2016.

[5] e-Estonia. 'e-identity solution'. Diponível: https://e-estonia.com/solutions/e-identity/id-card/. Acessado Maio 2019.

[6] Aadhaar. '*Unique Identification Authority of India*'. Disponível: https://uidai.gov.in. Acessado Julho 2019.

[7] NAKAMURA, E.T., RIBEIRO, S.L. '*Context-Based Blockchain Platform Definition and Analysys Methodology*'. The 18th International Conference on Security and Management (SAM19), Las Vegas, United States, July 2019.

[8] Cai, W., Wang, Z., Ernst, J., B., Hong, Z., Feng, C., Leung, V. C. M. '*Decentralized Applications: The Blockchain Empowered Software System*'. Disponível: https://www.researchgate.net/publication/327711685_Decentralized_Applications_The_Blockchain-Empowered_Software_System. Acessado Junho 2019.

[9] BlockchainHub. '*Blockchains & Distributed Ledger Technologies*'. Disponível: https://blockchainhub.net/blockchains-and-distributed-ledger-technologies-in-general/. Acessado Junho 2019.

[10] Ethereum. '*How to Build a Democracy on the Blockchain*'. Disponível: https://www.ethereum.org/dao. Acessado Junho 2019.

[11] Coindesk. '*Understand the DAO Attack*'. Disponível: http://www.coindesk.com/understanding-dao-hack-journalists/. Acessado Junho 2019.

[12] Bitcoin. 'Bitcoin Core'. Disponível: https://bitcoin.org/en/bitcoin-core/. Acessado Junho 2019.

[13] RIBEIRO, S. L., NAKAMURA, E. T. 'Context-Based Blockchain Platform Definition and Analysys Methodology – Results from the application in the BlockIoT Project'. International Conference on Advances in Cyber Security, Penang, Malaysia, 2019.

[14] TechTarget. '*Details emerging on Dyn DNS DDoS attack, Mirai IoT botnet*'. Disponível: http://searchsecurity.techtarget.com/news/450401962/Details-emerging-on-Dyn-DNS-DDoS-attack-Mirai-IoT-botnet. Acessado Maio 2019.

[15] The Security Ledger. '*Mirai, The Internet of Things Bot, Goes Open Source*'. Disponível: https://securityledger.com/2016/10/mirai-the-internet-of-things-bot-goes-open-source. Acessado Maio 2019.

[16] Imperva. 'Breaking Down Mirai: An IoT DDoS Botnet Analysis'. Disponível: https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html. Acessado Maio 2019.

[17] Dorri A.; Kanhere S.; Jurdak R.; Gauravaram P. '*Blockchain for IoT security and privacy: The case study of a smart home*'. IEEE. Disponível: http://ieeexplore.ieee.org/abstract/document/7917634. Acessado Maio 2019.

[18] Jun Zhou J.; Cao Z., Dong X.; Vasilakos A. '*Security and Privacy for Cloud-Based IoT: Challenges*'. IEEE. Disponível: http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=7823317. Acessado Maio 2019.

[19] Fremantle P.; Aziz B.; Kirkham T. '*Enhancing IoT Security and Privacy with Distributed Ledgers - a Position Paper*'. Maio 2019.

[20] BNDES. 'Internet das Coisas: Um plano de ação para o Brasil'. Disponível: https://www.bndes.gov.br/wps/portal/site/home/conhecimento/pesquisaedados/estudos/estudo-internet-das-coisas-iot/estudo-internet-das-coisas-um-plano-de-acao-para-o-brasil. Acessado Julho 2019.

[21] McKinsey. '*Blockchain beyond the hype: What is the strategic business value?*'. Disponível: https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/blockchain-beyond-the-hype-what-is-the-strategic-business-value. Acessado Junho 2019.

[22] Jianjun, S., Jiaqi, Y., Kem Z. K. '*Blockchain-based sharing services: What blockchain technology can contribute to smart cities*'. Financial Innovation, 2:26, DOI 10.1186/ s40854-016-0040-y. 2016.

[23] McKinsey. '*Using blockchain to improve data management in the public sector*'. Disponível: https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/using-blockchain-to-improve-data-management-in-the-public-sector. Acessado Maio 2019.

[24] OWI. '*Blockchain and Identity in 2018: A Year of Promise and Pilots*'. Disponível: https://oneworldidentity.com/research/blockchain-identity-2018-year-of-promise-pilots/. Acessado Julho de 2019.

[25] Peter Steiner. '*On the Internet nobody knows you are a dog*'. Disponível: https://en.wikipedia.org/wiki/On_the_Internet,_nobody_knows_you%27re_a_dog. Acessado Julho de 2019.

[26] Cameron, K. '*The Laws of Identity*'. Microsoft Corporation. Nov 2005. Disponível: http://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf. Acessado Julho 2019.

[27] Tobin, A., Reed, D. '*The Inevitable Rise of Self-Sovereign Identity*'. The Sovrin Foundation. March 2017. Disponível: https://sovrin.org/wp-content/uploads/2018/03/The-Inevitable-Rise-of-Self-Sovereign-Identity.pdf. Acessado Julho 2019.

[28] Lundkvist, C., Heck, R., Torstensson J., Mitton, Z., Sena, M. '*uPort: A Platform for Self-Sovereign Identity*'. Feb 2017. Disponível: http://blockchainlab.com/pdf/uPort_whitepaper_DRAFT20161020.pdf. Acessado: Julho 2019.

[29] Ali, M., Nelson, J., Shea, R., Freedman, M. J. '*Blockstack: A Global Naming and Storage System Secured by Blockchains*'. 2016 USENIX *Annual Technical Conference* (USENIX ATC 16), Denver, CO, 2016, pp. 181–194. Disponível: https://www.usenix.org/node/196209. Acessado Julho 2019.

[30] The White House. '*National Strategy for Trusted Identities in Cyberspace: Enhancing Online Choice, Efficiency, Security, and Privacy*', Apr 2011. Disponível: https://www.hsdl.org/?view&did=7010. Acessado Julho 2019.

[31] United Nations. '*Transforming our world: the 2030 agenda for sustainable development*'. Sep 2015. Disponível: https://www.unfpa.org/resources/transforming-our-world-2030-agenda-sustainable-development. Acessado Julho 2019.

[32] Nakamoto, S. '*A Peer-to-Peer Electronic Cash System*'. Disponível: www.bitcoin.org/bitcoin.pdf. Acessado Maio 2019.

[33] ISO/IEC. '*ISO/IEC 24760-1:2019 – Information technology – Security techniques – A framework for identity management – Part1: Terminology and concepts*'. May 2019. Disponível: https://www.iso.org/standard/77582.html. Acessado Julho 2019.

[34] Zooko, W. '*Names: Distributed, Secure, Human-Readable: Choose Two*'. May 2017. Disponível: http://www.cs.princeton.edu/courses/archive/spr17/cos518/papers/zooko-triangle.pdf. Acessado Junho 2019.

[35] Hyperledger. '*Hyperledger Indy*'. Disponível: https://www.hyperledger.org/projects/hyperledger-indy. Acessado Maio 2019.

[36] W3C – DID. '*Decentralized Identifiers (DIDs)*'. Disponível: https://w3c-ccg.github.io/did-spec/. Acessado Maio 2019.

[37] Aublin, P. L., Mokhtar, S., B., Quéma, V. '*RBFT: Redundant Byzantine Fault Tolerance*'. Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on, 2013, pp. 297–306. Disponível: https://pakupaku.me/plaublin/rbft/5000a297.pdf. Acessado Maio 2019.

[38] Dunphy, P., Petitcolas, F., A., P. '*A First Look at Identity Management Schemes on the Blockchain*'. IEEE Security and Privacy Magazine. 2018.

[39] Sovrin. '*Technical Architecture Diagrams*'. Disponível: https://forum.sovrin.org/t/technical-architecture-diagrams/62/3. Acessado Junho 2019.

[40] Ethereum. '*A Next-Generation Smart Contract and Decentralized Application Platform*'. Disponível: https://github.com/ethereum/wiki/wiki/White-Paper. Acessado Maio 2019.

[41] ShoCard SITA. '*Travel Identity of the Future – White Paper*'. 2016. Disponível: https://shocard.com/wp-content/uploads/2016/11/travel-identity-of-the-future.pdf. Acessado Junho 2019.

[42] BENNET, Colin. Regulating privacy: data protection and public policy in Europe and United States. Ithaca, New York: Cornell University Press, 1992

[43] COTS, Márcio, Oliveira, Ricardo. Lei Geral de Proteção de Dados Pessoais Comentada. 1ª. Edição. São Paulo: Thomson Reuters Brasil, 2018.

[44] W3C. Decentralized Identifiers (DIDs) v0.13. Disponível em: <https://w3c-ccg.github.io/did-spec/>. Acesso em: 09/08/2019.

[45] Sovrin e Evernym. What Goes on the Ledger? Disponível em: <https://sovrin.org/wp-content/uploads/2018/10/What-Goes-On-The-Ledger.pdf>. Acesso em: 02/02/2019.

[46] LEORATTI, Alexandre. Para ministro do STJ, LGPD gera 'mais dúvidas do que certezas'. Jota, 11/12/2018. Disponível em: <https://www.jota.info/justica/lgpd-revisao-jurisprudencia-stj-11122018>. Acesso em: 02/02/2019.

[47] ZYSKING, Guy et al, 'Decentralizing Privacy: Using Blockchain to Protect Personal Data' (2015) IEEE Security and Privacy Workshops.

[48] REED, Drummond. The Story of SSI Open Standards Background on the Foundation of Self Sovereign Identity: DIDs, DKMS, DID Auth and Verifiable Credentials. 26 April 2018 SSIMeetup.org

Disponível em: <https://ssimeetup.org/story-open-ssi-standards-drummond-reed-evernym-webinar-1/>. Acesso em:092/08/2019.

[49] W3C Community Group. Decentralized Identifiers (DIDs) v0.13 - Data Model and Syntaxes. Disponível em: <https://w3c-ccg.github.io/did-spec/#introduction>. Acesso em: 09/08/2019.

[50] Hyperledger Indy. DKMS (Decentralized Key Management System) Design and Architecture V3. Disponível em: <https://github.com/hyperledger/indy-sdk/blob/677a0439487a1b7ce64c2e62671ed3e0079cc11f/doc/design/005-dkms/DKMS%20Design%20and%20Architecture%20V3.md>. Acesso em: 09/08/2019.

[51] ABNT/CEE-307. Blockchain e tecnologias de registro distribuídas – Conceitos e elementos da tecnologia Blockchain – Parte 6: Segurança, privacidade e identidade. Disponível em: <https://isolutions.iso.org/ecom/livelink/fetch/-54235805/54235807/54250561/70060171/P_307.000.000-001_Parte_06_Ago19.pdf?nodeid=70043683&vernum=-2>. Acesso em 09/08/2019.

[52] Rolf Oppliger. Microsoft .NET Passport and identity management. Information Security Technical Report, 2004.

[53] Kormann D, Rubin A. Risks of the passport single signon protocol. IEEE Computer Networks 2000.

Disponível em <https://www.cs.jhu.edu/~rubin/courses/sp03/papers/passport.pdf>. Acesso em 10/09/2019.

[54] O'Donnell, The Current and Future State of Digital Wallets. 1ª. Edição. Canadá: Creative Commons, 2019.

[55] Dan Boneh; Ben Lynn & Hovav Shacham (2004). "Short Signatures from the Weil Pairing". Journal of Cryptology.

[56] Dan Boneh, Matthew K. Franklin, Identity-Based Encryption from the Weil Pairing, SIAM J. of Computing, Vol. 32, 2003

[57] N. Theriault. Index calculus attack for hyperelliptic curves of small genus, 2003.

Disponível em <https://www.iacr.org/archive/asiacrypt2003/02_Session02/19_056/28940307.pdf>. Acesso em 10/08/2019.

[58] J-S Coron. On the Exact Security of Full Domain Hash. CRYPTO 2000. Disponível em <https://www.iacr.org/archive/crypto2000/18800229/18800229.pdf>. Acesso em 10/09/2019.

[59] D. Boneh, C. Gentry, H. Shacham, B. Lynn. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In proceedings of Eurocrypt 2003. Disponível em <https://crypto.stanford.edu/~dabo/pubs/papers/aggreg.pdf>. Acesso em 10/09/2019.

[60] Camenisch J., Lysyanskaya A. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In: Yung M. (eds) Advances in Cryptology — CRYPTO 2002. Disponível em <https://link.springer.com/content/pdf/10.1007/3-540-45708-9_5.pdf>. Acesso em 10/09/2019.